



دانشگاه شمال(آمل)

## دانشکده فنی مهندسی - گروه کامپیووتر پایان نامه دوره کارشناسی

موضوع:

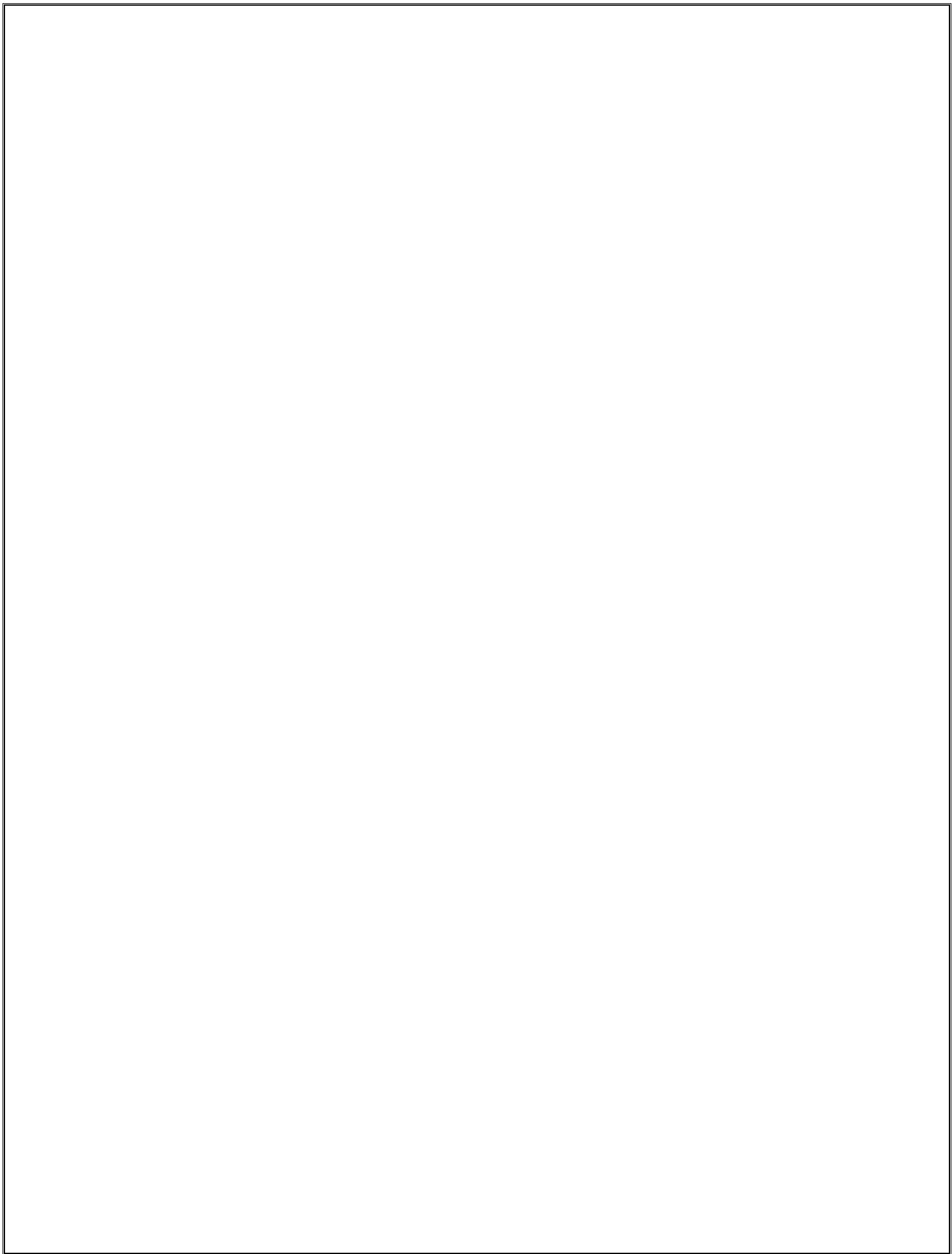
# پردازش تصویر فازی

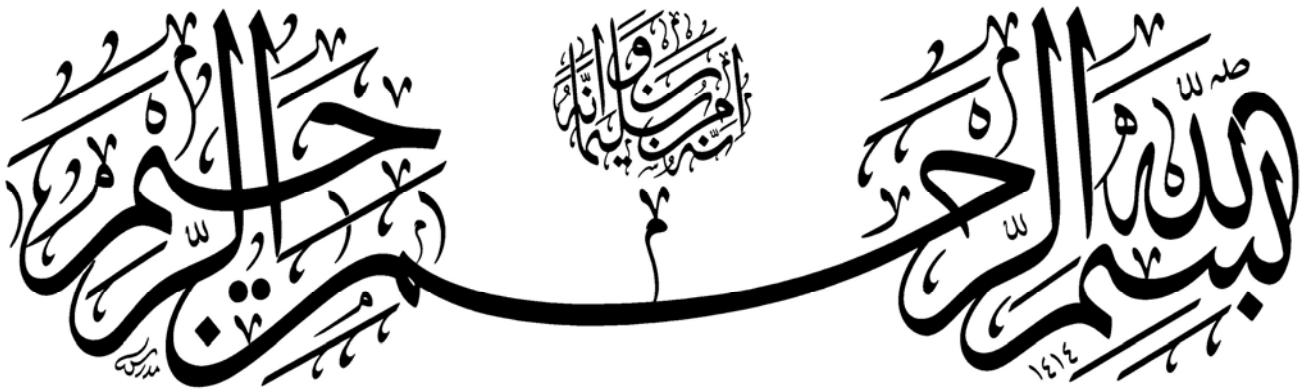
استاد راهنما:

جناب آقای مهندس محمد روحانی

نگارش:

معین نبی





به نام خداوند عشق که همه چیز است.

هر کس روزنه‌ای ست به سوی خداوند اگر اندوهناک شود،

اگر بشدت اندوهناک شود...

"قرآن"

با ژرف ترین سپاس ها:

از لطف بی پایان الهی که آموختن را می آموزد.

از قلب پاک مادرم و دست های مقدس پدرم که سرچشمه‌ی لذت از تمام موهبت های الهی است.

از اساتید محترم دانشکده فنی و مهندسی دانشگاه شمال، که رهنمون هایشان همیشه رهگشا بهم بود.

و از همه دوستانم که مرا در بهتر نگریستن و اندیشیدن یاری داده اند.

معین نبی

تقدیم به پدر و مادرم؛

آنان که تصویرشان در ذهنم، بی نیاز از هر ارتقایی است...

## **چکیده پایان نامه (دیباچه)**

در تحقیق حاضر، ابتدا به بیان نظریه فازی پرداخته سپس با توضیح مقدماتی از پردازش تصویر، کاربردهای این نظریه را در علم پردازش تصویر بیان می‌کنیم.

در فصول این تحقیق به موضوعات زیر می‌پردازیم:

### **فصل 1: نظریه فازی**

#### **Fuzzy Theory**

در این فصل ابتدا به پیشینه فلسفی و تاریخچه تئوری فازی پرداخته، سپس با معرفی این نظریه، توضیح کوتاهی پیرامون منطق فازی، مجموعه‌های فازی، اعداد فازی، کنترل فازی داده و کاربرد آنرا در پردازش تصویر بیان می‌کنیم.

### **فصل 2: پردازش تصویر:**

#### **Image Processing:**

در این فصل ابتدا به بیان مفاهیم پایه‌ای در پردازش تصویر دیجیتالی پرداخته و با ارائه مثال‌هایی عملیات روی تصویر و هیستوگرام را بطور مختصر توضیح می‌دهیم. سپس ارتقای تصویر و فیلترهای حوزه مکان را بررسی کرده و در انتها ارتقای تصویر در حوزه فرکانسی را مختصراً توضیح می‌دهیم.

### **فصل 3: پردازش تصویر فازی :**

#### **Fuzzy Image Processing :**

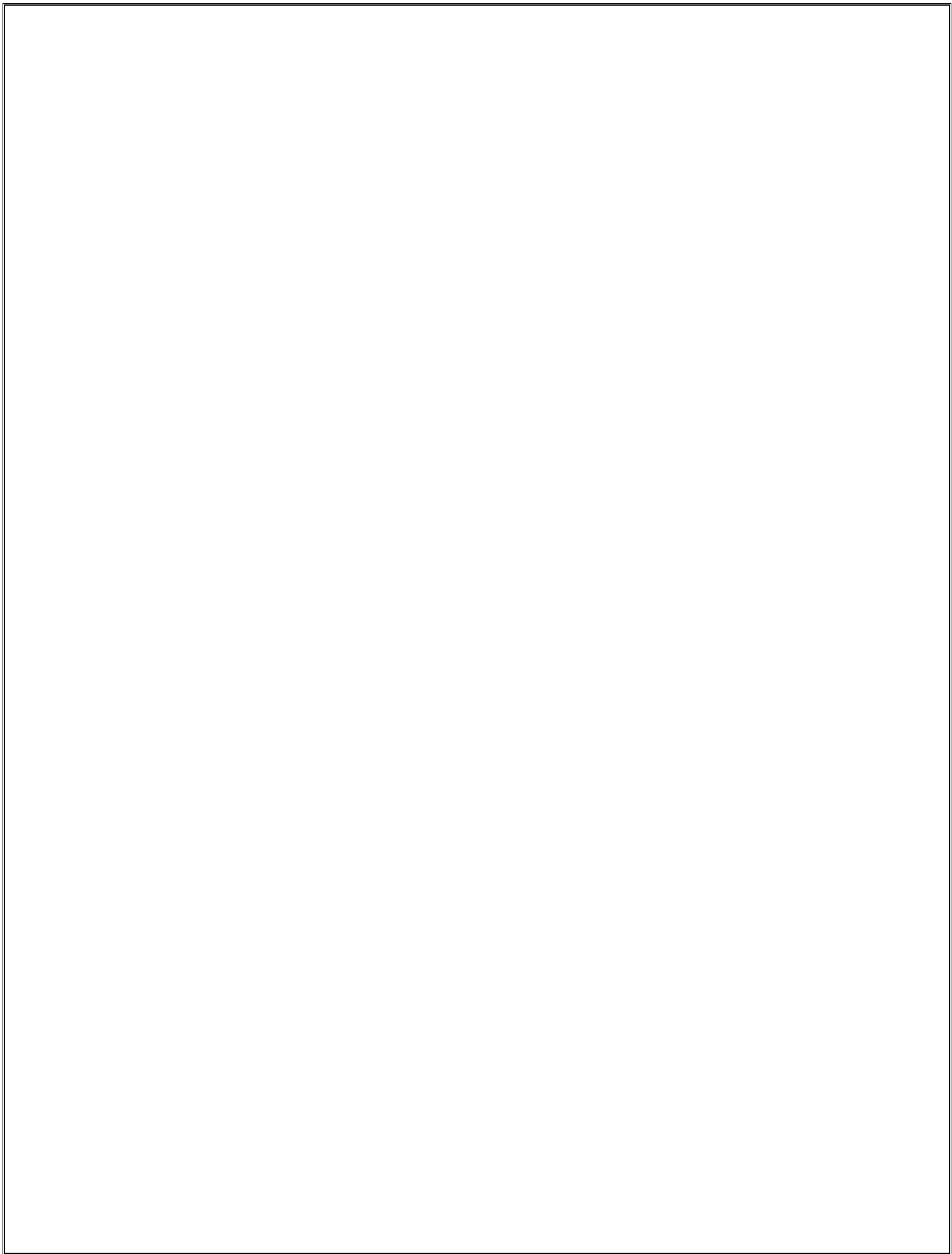
در این فصل نحوه‌ی به کار گیری منطق فازی در پردازش تصویر را توضیح داده و تئوری پردازش تصویر فازی در هندسه فازی را مورد بررسی قرار می‌دهیم. سپس تشخیص لبه فازی (Fuzzy Edge Detection)، بهبود تصویر فازی (Fuzzy Image Enhancement) و بخش‌بندی تصویر فازی (Fuzzy Image Segmentation) را توضیح می‌دهیم.

### **پیوست ها**

- 1 . نحوه بکارگیری از نرم افزار Matlab در پردازش تصویر با استفاده از Image Processing Toolbox
- 2 . مقاله‌ای با عنوان

RECOGNITION OF PERSIAN ONLINE HANDWRITING USING ELASTIC FUZZY PATTERN RECOGNITION

نوشته رامین حلاوتی و سعید باقری پیرامون کاربرد پردازش تصویر فازی



فصل اول

# منطق فازی

## مقدمه:

حتماً بارها شنیده‌اید که کامپیوتر از یک منطق صفر و یک تبعیت می‌کند. در چارچوب این منطق، چیزها یا درستند یا نادرست، وجود دارند یا ندارند. اما اینیشتین می‌گوید: <آن جایی که قوانین ریاضیات (کلاسیک) به واقعیات مربوط می‌شوند، مطمئن نیستند و آنجا که آن‌ها مطمئن هستند، نمی‌توانند به واقعیت اشاره داشته باشند.> هنگامی که درباره درستی یا نادرستی پدیده‌ها و اشیایی صحبت می‌کنیم که در دنیای واقعی با آن‌ها سروکار داریم، توصیف اینیشتین تجسمی است از ناکارآمدی قوانین منطق کلاسیک در علم ریاضیات. از این رو می‌بینیم اندیشه نسبیت شکل می‌گیرد و توسعه می‌یابد. در اینجا می‌خواهیم به اختصار با منطق فازی آشنا شویم. منطقی که دنیا را نه به صورت حقایق صفر و یکی، بلکه به صورت طیفی خاکستری از واقعیت‌ها می‌بیند و در هوش مصنوعی کاربرد فراوانی یافته‌است.

### کجا اتومبیل خود را پارک می‌کنید؟

تصور کنید یک روز مطلع می‌شوید، نمایشگاه پوشکی در گوشه‌ای از شهر برپا شده است و تصمیم می‌گیرید، یک روز عصر به اتفاق خانواده سری به این نمایشگاه بزند. چون محل نمایشگاه کمی دور است، از اتومبیل استفاده می‌کنید، اما وقتی به محل نمایشگاه می‌رسید، متوجه می‌شود که عده زیادی به آنجا آمده‌اند و پارکینگ نمایشگاه تا چشم کار می‌کند، پر شده است.

اما چون حوصله صرف وقت برای پیدا کردن محل دیگری جهت پارک اتومبیل ندارید، با خود می‌گویید: <هر طور شده باید جای پارکی در این پارکینگ پیدا کنم.> سرانجام در گوشه‌ای از این پارکینگ محلی را پیدا می‌کنید که یک ماشین به طور کامل در آن جا نمی‌شود، اما با کمی اغماض می‌شود یک ماشین را در آن جای داد، هرچند که این ریسک وجود دارد که فضای عبور و مرور دیگر خودروها را تنگ کنید و آن‌ها هنگام حرکت به خودرو شما آسیب برسانند. اما به هر حال تصمیم می‌گیرید و ماشین خود را پارک می‌کنید.

بسیار خوب! اکنون بباید بررسی کنیم شما دقیقاً چه کار کردید؟ شما دنبال جای توقف یک اتومبیل می‌گشتبید. آیا پیدا کردید؟ هم بله، هم نه. شما در ابتداء می‌خواستید ماشین را در جای مناسبی پارک کنید. آیا چنین عملی انجام دادید؟ از یک نظر بله، از یک دیدگاه نه. در مقایسه با وقت و انرژی لازم برای پیدا کردن یک مکان راحت برای توقف خودرو، شما جای مناسبی پیدا کردید. چون ممکن بود تا شب دنبال جا بگردید و چنین جایی را پیدا نکنید. اما این نظر که اتومبیل را در جایی پارک کردید که فضای کافی برای قرارگرفتن ماشین شما نداشت، نمی‌توان گفت جای مناسبی است.

اگر به منطق کلاسیک در علم ریاضیات مراجعه کنیم و این پرسش را مطرح نماییم که قبل از ورود به پارکینگ چند درصد احتمال می‌دادید جایی برای پارک کردن پیدا کنید، پاسخ بستگی به این دارد که واقعاً چه تعداد مکان مناسب (فضای کافی) برای توقف خودروها در آنجا وجود داشت؟ اگر به حافظه خود رجوع کنید، شاید به یاد بیاورید که هنگام ورود به پارکینگ و چرخیدن در قسمت‌های مختلف آن، گاهی خودروهایی را می‌دیدید که طوری پارک کرده‌اند که مکان یک و نیم خودرو را اشغال کرده‌اند. بعضی دیگر نیز کج و معوج پارک کرده بودند و این فکر از ذهن شما چندبار گذشت که اگر صاحب بعضی از این خودروها درست پارک کرده بودند، الان جای خالی برای پارک کردن چندین ماشین دیگر هم وجود داشت.

به این ترتیب علم ریاضیات و آمار و احتمال در مواجهه با چنین شرایطی قادر به پاسخگویی نیست. اگر قرار بود بر اساس منطق صفر و یک یا باینری کامپیوتر، روابطی ساخته شود تا اتومبیل شما را در یک مکان مناسب پارک کند، احتمالش کم بود. چنین روابطی به احتمال زیاد ناکام از پارکینگ خارج می‌شد. پس شما با چه منطقی توانستید اتومبیل خود را پارک کنید؟ شما از منطق فازی استفاده کردید.

## دنیای فازی

می‌پرسم <هوا ابری است یا آفتای؟> پاسخ می‌دهی: نیمه‌ابری. می‌پرسم <آیا همه آنچه که دیروز به من گفتی، راست بود؟> پاسخ می‌دهی: بیشتر آن حقیقت داشت. ما در زندگی روزمره بارها از منطق فازی استفاده می‌کنیم. واقعیت این است که دنیای صفر و یک، دنیایی انتزاعی و خیالی است. به ندرت پیش می‌آید موضوعی صدرصد درست یا صدرصد نادرست باشد؛ زیرا در دنیای واقعی در بسیاری از مواقع، همه‌چیز منظم و مرتب سرجایش نیست.

از نخستین روز تولد اندیشه فازی، بیش از چهل سال می‌گذرد. در این مدت نظریه فازی، چارچوب فکری و علمی جدیدی را در محافل آکادمیک و مهندسی معرفی نموده و دیدگاه دانشمندان را نسبت به کم و کیف دنیای اطراف ما تغییر داده است. منطق فازی جهان‌بینی بدیع و واقع‌گرایانه‌ای است که به اصلاح شالوده منطق علمی و ذهنی بشر کمک شایانی کرده است.

## پیشینه فلسفی:

### • ریشه‌های تفکر فازی

تفکر فازی دیدگاه تازه‌ای را معرفی می‌کند که تعمیم منطق ارسطوی است. این تفکر، از فلسفه نشات می‌گیرد که سابقه چند هزار ساله و به قدمت فلسفه تاریخ دارد. همان‌گونه که فلسفه ادیان الهی با طبیعت و سرشت انسان سازگار است، تفکر فازی با الهام از فلسفه شرقی، جهان را همان‌گونه که هست معرفی می‌کند.

بودا نیم قرن قبل از میلاد مسیح و تقریباً دو قرن قبل از ارسطو زندگی می‌کرد. اولین قدم در سیستم اعتقادی وی گریز از جهان کلمات سیاه و سفید و درین این جهان دو ارزشی و هدفش نگریستن به جهان به صورتی فی نفسه است. باید جهان را سراسر تناقض دید. جهانی که در آن چیزها و ناچیزها وجود دارد.

حتی هنگامی که پارامیندس حدود 400 سال قبل از میلاد مسیح اولین نسخه این قانون را بیان کرد، با مخالفت‌های شدید مواجه شد. همچنین هراکلیتوس پیش از او بیان کرد که هرچیزی می‌تواند هم زمان درست و یا نادرست باشد. در تاریخ آمده است که یونانیان باستان عقیده داشتند موجوداتی اساطیری به نام "ابوالهول" در جاده‌ها و راه‌ها کمین می‌کنند و مسافران تنها را می‌آزارند. ابوالهول‌ها موجودات غول پیکری بودند، ترکیبی از انسان و شیر، چیزی شبیه ابوالهول مصر، که در جاده‌ها می‌ایستادند و راه مسافران را می‌بستند. آن‌ها از مسافران یک سوال می‌پرسیدند، اگر جواب درست می‌دادند اجازه عبور می‌یافتند و گرنه ابوالهول آنها را از هم می‌دربندند.

ادیپوس شهریار در نمایشنامه سوفوکل با پاسخ دادن به سوال یک ابوالهول، شهری را از مصیبت نجات می‌دهد و به پادشاهی می‌رسد. شکل‌گیری چنین اسطوره زیبایی در بیش از دو هزار سال پیش تنها یک دلیل می‌تواند داشته باشد: یونانیان، دانایی را تنها راه زیستن یا حتی شهریاری می‌دانستند. دانایی یگانه راه عبور از جاده هاست. مسافر خاک آلود با ترس و لرز می‌پرسد: آگاهی به چه قیمت؟ و ابوالهول با خونسردی جواب می‌دهد: به قیمت زندگی ات!! اما آگاهی، چیزی دست نیافتدی است. پاسخ دادن به سوال‌های ابوالهول کار چندان آسانی نیست و حتی غیر ممکن است.

ابوالهول با پنجه‌ی غول پیکرش راه را بر تو می‌بندد و از تو می‌پرسد: اگر یک مرد بگوید من دروغ می‌گویم، آیا باز هم دروغ می‌گوید؟ پاسخ مثبت یا منفی به این سوال بی‌فاایده است. اگر ابوالهول چنین سوالی از شما بپرسد، بهتر است سکوت کنید و خودتان را برای مرگ آماده نمایید. معماهای بی‌جواب ابوالهول به **SOROS** در زبان یونانی به معنای خرمن است. چرا که معنای خرمن، لایحل ترین سوال ابوالهول به حساب می‌آید.

ابوالهول: آیا یک دانه گندم می‌تواند به تنها ی خرمنی بسازد؟ دو گندم چطور؟ سه دانه گندم چطور؟

ابوالهول این سوال های سلسله‌وار را از تو می‌پرسد و پاسخ تو به هر کدام از آنها یک "نه" قاطع خواهد بود. احتمالاً یک شبانه روزی می‌گذرد و تو وقتی به خود می‌آیی که او می‌پرسد: آیا هزار دانه گندم خرمنی تشکیل می‌دهد؟ آن وقت است که تو بازی را باخته‌ای بی‌آنکه مرز تبدیل دانه به خرمن را دانسته باشی. فرض کنید در آخرین لحظه جرات می‌یابی و به ابوالهول می‌گویی: بله... ده هزار گندم یک خرمن است، در این صورت ابوالهول سولاتش را این گونه پی‌می‌گیرد:

"اگر ده هزار دانه گندم خرمنی بسازد، پس نه هزار و نهصد و نود و نه گندم هم خرمنی می‌سازد؟"

او اعداد را یکی یکی پایین می‌آورد تا به صفر برسد و آن وقت است که تو می‌توانی آن اعداد را شمارش معکوس زندگی ات به حساب آوری. معماهای ابوالهول به تو نشان می‌دهد ساده ترین مفاهیمی که فکر می‌کنی به آن آگاهی داری در واقع تاریک ترین نقاط فکر تو هستند. همه ما ادعا می‌کنیم تفاوت یک دانه گندم و خرمن را می‌دانیم اما ابوالهول به ما ثابت می‌کند که ما هیچ نمی‌دانیم.

فلسفه ابهام در یونان باستان بازار گرمی داشت. روزگاری که افلاطون، ارسطو و شاگردانش از منطق و دانایی حرف می‌زنند، کسان دیگری هم بودند که در کوچه‌های آتن پرسه می‌زنند و به رهگذری یاد آوری می‌کرند که هیچ حقیقتی وجود ندارد. آن‌ها در اصطلاح " Sofiست " بودند. فیلسوفانی که در فلسفه اسلامی به سفسطه گرایان یا " سفسطائیان " مشهور شدند. Sofiست‌ها این توانایی را داشتند که به شیوه ابوالهول‌ها با چند بازی به ما بفهمانند که تا امروز اشتباه می‌کردیم. روز لزوماً روشن‌تر از شب نیست. دو خط موازی ممکن است روزی بهم برسند. قدرت گرفتن فلسفه افلاطونی و ارسطوی باعث شد نسل Sofiست‌ها در یونان باستان منقرض شود. اما فلسفه زمان امروز کم کم به این نتیجه می‌رسد که اگر در تاریخ فلسفه، امثال افلاطون و ارسطو راه Sofiست‌ها را سد نکرده بودند، بشر به افق‌های اندیشه بازتری راه یافته بود.

یکی از نظریه‌هایی که ارسطو بنا نهاد قانونی به نام " اصل عدم شمول میانی " بود. بر اساس این نظریه، هر گزاره‌ی قیاسی دارای ارزش " درستی " (true) یا نادرستی (false) است و هیچ مقدار سومی وجود ندارد که گزاره بتواند آن را بگیرد. به همین دلیل به آن اصل عدم شمول میانی می‌گویند زیرا بر اساس این نظریه چیزی بین درست و نادرست وجود ندارد! اما افلاطون بود که چیزی را که امروزه با نام منطق فازی می‌شناسیم پایه‌ریزی کرد. او نشان داد که ناحیه سومی هم آنسوی درست و نادرست وجود دارد که گزاره‌های قیاسی می‌توانند مقادیر آن را بگیرند.

## • اصل تضاد در منطق دیالکتیک هگل (قرن نوزدهم)

هگل فیلسوف آلمانی معتقد است هر نقطه از مسیر حرکت است، هم هست و هم نیست. مثلاً هر کسی در هر لحظه از عمر خود کسی است که قبلاً بوده است و هم کسی است که نبوده است. زیرا اگر او همان باشد که بوده است، دیگر تغییر نکرده است، و اگر او همان نباشد که بوده است، در آن لحظه از هیچ به وجود آمده است.

اصل تضاد امور، اساس منطق دیالکتیک هگل است. وی معتقد است فلسفه من، فلسفه جهان تمدن است و منطق صوری، بخشی از آن است.

## • فلسفه ابهام و تضاد در قرن بیستم

در میان فیلسوفان زمان قرن بیستم، پیتر گیج تقریباً اولین کسی بود که بحث فلسفه ابهام را باز دیگر زنده کرد. مثال‌هایی که او برای اثبات فلسفه اش بکار می‌برد بی‌شباهت به معماهای ابوالهول و Sofiست‌ها نبود. آیا یک بچه قورباغه که هنوز در آب باله هایش شنا می‌کند، می‌تواند در یک ثانیه تبدیل به قورباغه شود؟ مسلمان جواب فلسفه ارسطوی منفی است. اما اگر بپذیریم بچه غورباغه دو ثانیه بعد هنوز یک بچه غورباغه است، پس به راحتی می‌توانیم پیش بینی کنیم که او در دو ثانیه بعد هم تبدیل به قورباغه نخواهد شد. اگر دو ثانیه بعد او هنوز بچه قورباغه باشد، در ثانیه سوم هم به قورباغه تبدیل نمی‌شود، چرا که یک ثانیه برای قورباغه شدن هنوز مدت بسیار کمی است. اگر این رشتہ را ادامه دهی تنها به یک نتیجه می‌رسیم: " بچه قورباغه‌ی

بیچاره، هیچ وقت قورباغه نخواهد شد!!! "پیتر گیج در فلسفه خود می گوید: چیزی به نام هویت مطلق وجود ندارد. هویت ما و همه چیزهای اطرافمان تنها به طور نسبی تعریف می شوند، یعنی نسبت به نقطه قبل و نقطه بعد. اگر با فلسفه ارسطوی بخواهی به ابوالهولها جواب بدھی، مسلماً خودت را خراب کرده ای. در منطق ارسطو باید در جواب هرسوال یا بگویی بلی یا بگویی نه و این دقیقاً همان چیزی است که سوال کننده انتظار دارد تا تو را به پرتوگاه هل دهد. اما در فلسفه سوفیست ها ی باستان، در جواب هیچ سوالی، نمی توان جوابی داد. جواب یک سوال می تواند امروز بله باشد، فردا نه. جواب یک سوال می تواند در هر لحظه هم مثبت باشد و هم منفی.

#### • منطق چند ارزشی در قرن بیستم

در حدود سال 1920 برای نخستین بار لوکازی اویچ هنگامی که منطق سه - ارزشی را با همراهی ریاضیات توصیف نمود، یک راهکار سیستماتیک برای منطق دو - ارزشی ارسطو بیان کرد. بعدها وی منطق چهار - ارزشی و پنج - ارزشی کشف کرد و سپس اعلام کرد که اساساً هیچ محدودیتی در منطق بی نهایت - ارزشی وجود ندارد. لوکازی اویچ نتیجه گیری کرد که منطق سه و بی نهایت - ارزشی، علی رغم این که جاذبه زیادی داشت، اما براساس یک منطق چهار - ارزشی استوار بود که به نظر می رسید بیشترین سازگاری را با منطق ارسطوی داشته باشد. همچینین لازم به ذکر است که نات یک منطق سه - ارزشی را مشابه با منطق لوکازی اویچ پشنهد نمود، زیرا وی معتقد بود ریاضیات باید به مراتب فراتر از منطق دو - ارزشی سنتی باشد.

ارزش‌های اول و دوم همان درست و نادرست و ارزش سوم را "possible" یا ممکن نام نهاد. این ارزش سوم طبق تعریف او چیزی بین true و false بود.

در مورد منطق سه ارزشی داستانی را نقل می کنند که بیشتر به عنوان یک پارادوکس برای آن دسته از منطق دانانی که می گفتند منطق سه ارزشی چیزی غیر ضروری و یک تجمل است به کار می رود :

نقل می کنند که یک کشتی در اقیانوس دچار طوفان می شود و کشتی غرق می شود یک نفر منطق دان که جزء مسافران کشتی بود نجات می یابد و با شنا کردن خود را به یک جزیره می رساند.

در آن جزیره از فرط خستگی به خواب می رود موقعی که بیدار می شود می بیند که در یک ساختمان مجلل است و جمعیتی وی را احاطه کرده اند . می پرسد که جریان چیست و من کجا هستم ؟ یکی از آنان جواب می دهد که اینجا دادگاه شهر ماست و تو اسیر ما هستی ؟

می گوید من که با پای خودم به این جزیره آمده ام و جنگی هم بین ما در نگرفته است پس چرا من اسیر شما هستم؟ جواب میدهند در این جزیره رسم است که هر سال ما اهالی شهر، در یک روز معین به ساحل می رویم و اولین غریبه ای را که می بینیم اسیر می کنیم و به شهر می آوریم و سپس از او سوالی می پرسیم. ما دو مکان برای کشتن افراد داریم یک مکان به نام مکان true یا درست مشهور است و مکان دیگر هم به نام false یا نادرست . پس از آنکه اسیر ما سوال را جواب داد قاضی این دادگاه جواب را بررسی می کند و درست یا نادرست بودن جواب را تشخیص می دهد و اسیر را به یکی از دو مکان درست یا نادرست می فرستد در هر دوی این مکان ها هم یک قاضی هست که او هم پس از شنیدن ماجرا و جوابی که اسیر داده است حکم کشتن اسیر را می دهد. هر گاه قاضی اول و یکی از دو قاضی در مورد درست یا نادرست بودن جواب اسیر موافق باشند اسیر کشته می شود . حال تو هم باید به سوال ما جواب بدھی !

حاکم این سوال را می پرسد : "آیا می دانی که به زودی کشته خواهی شد؟" منطق دان هم فوراً جواب می دهد : "بلی! من را در مکان false خواهید کشت!"

حاکم فوراً دستور می دهد که او جواب من را درست داده است او را به مکان true ببرید. منطق دان را به مکان true قاضی آنجا پس از شنیدن دو باره جواب اسیر از زبان مامور همراه او می گوید او دروغ گفته است چون اینجا مکان true است و او گفته است که او را در مکان false خواهیم کشت او را به بخش false ببرید چون دروغ گفته است.

در مکان false هم قاضی پس از شنیدن ماجرا می گوید که او راست گفته است او را به بخش true ببرید چون او راست گفته است . اینجا بخش false است و او گفته است که در بخش false کشته خواهد شد .

منطق دان کاری کرد که در یک دور تسلسل بین بخش های true ، false گرفتار شود و چون راهکاری برای کشتن او نداشتند عاقبت از کشتن او صرفنظر کردند.

## • پیدایش فازی

اولین بار نظریه مجموعه های فازی در سال 1965 توسط پروفسور لطفی عسکرزاده دانشمند ایرانی تبار و استاد دانشگاه برکلی، عرضه شد. خلق، بسط و توسعه اندیشه فازی توسط وی طی سالیان درازی صورت گرفته است. برگردیم به این که چه طوری شد که این ذهنیت در فکر آقای زاده شکل گرفت و جرقه زد حکایت جالبی دارد. یکی از شاگردان ایشان به اسم آقای باردکوسکو در یک کتاب تحت عنوان *فازتینکینگ* که خوشبختانه این کتاب به فارسی هم ترجمه شده نقل می کند از بیان ایشان که در دانشگاه بریکلی آقای زاده با یکی از اساتید به نام آن حوزه در یک آزمایشگاهی کار تحقیقاتی انجام می دهند و اینها به صورت مشترک این کار را جلو بردند تا منجر به دست آورد قطعی شد و یک فیلترها و یا صافی هایی را استفاده کردند و طراحی کردند و ساختند که این فیلترها قادر بود که هر نوع نویز یا پارازیت هایی را که دارای توزیع رئوسی هستند فیلتر کند. مجری: در واقع مدارهای الکترونیکی یا فیلترهای نوری بود. دکتر ستایشی: بله صافی الکترونیکی بود که اگر چنان چه سیگنالی حاوی نویز با توزیع رئوسی بود این می توانست فیلتر کند و این کاربری خیلی عمدی ای در دنیا دارد در هوایپیما و موشک و غیره. اتفاقی که اینجا می افتد این است که این فیلتر به دنیا معرفی شود و این دستگاه و وسیله در محیط طرح موضوع آقای زاده از این همکارش یک مقدار دیرتر می رسد و وقتی که می رسد می بیند که این دوست ایشان شروع کردند به معرفی این فیلتر و صافی و در واقع اسمی هم از آقای زاده بره نمی شود، ایشان نقل می کند که من در انتهای سالن نشسته بودم و به این کار عجیب فکر می کردم که بعد از چند سال کار اساسی و زیر بنایی در حوزه مهندسی الکترونیک و برق چه طور می شود که علی رغم مشارکت جدی که من در این کار داشتم یک باره هیچ اسمی از من بره نمی شود. خوب حق داشته آن موقع هر کاری را انجام بدهد و بلند سود و اعتراضی کند و اما خیلی ساكت تا انتهای جلسه می نشیند و وقتی که صحبت آن دوستش تمام می شود و جلسه دارد ختم می شود، اولین نفری است که از سالن خارج می شود ایشان نقل می کند که در آن زمان من فکر می کردم که کار ایشان را چگونه جبران کنم و چگونه می توانم کاری کنم که نه تنها کاری را که این صافی انجام می دهد بلکه در ابعاد وسیع تر بتواند عمدۀ نویزها را با هر توزیعی صاف کند و یا فیلتر کند یک خبرنگار به آن می رسد، می گوید آقای زاده نظر شما راجع به این فیلتر چیست و ایشان در بیان زبان انگلیسی که می تواند از واژه خیلی خوب یا خیلی بد سو یا تو استفاده کند می گوید: دیس فیلتر ایز تو گوسین یعنی می گوید که این فیلتر خیلی گوسی است و به یک ترتیبی عنوان می کند که این از یک محدودیت گوسی بودن نویز برخوردار است، کاملاً وابسته به توزیع گو سی نویز است در حالی که این فیلتر فیلتر بسیار موفقی است، اما خودش نقل می کند که از همان لحظه که به آزمایشگاه برگشتم دنبال یک مفهومی رفتم که این مفهوم بتواند به صورت عام و جامع تمام پدیده هایی که از هر نوع توزیعی برخوردار هستند در بر بگیرد. لذا به جامعیت مجموعه های فازی دست پیدا می کند.

واژه فازی در فرهنگ لغت آکسفورد بصورت مبهم، گنگ، نا دقیق، گیج، مغشوش، در هم و نامشخص تعریف شده است. اساساً گرچه سیستم های فازی پدیده های فازی پدیده های غیر قطعی و نامشخص را توصیف می کنند، با این حال خود تئوری فازی یک تئوری دقیق می باشد. در این متن دو نوع توجیه برای تئوری سیستم های فازی وجود دارد:

• دنیای واقعی ما بسیار پیچیده تر از آن است که بتوان یک توصیف و تعریف دقیق برای آن بددست آورد بنابراین باید یک توصیف تقریبی یا همان فازی که قابل قبول و قابل تجزیه و تحلیل باشد. برای یک مدل معرفی شود.

• با حرکت ما بسوی عصر اطلاعات، دانش و معرفت بشری بسیار اهمیت پیدا می کند. بنابراین ما به فرضیه ای نیاز داریم که بتواند دانش بشری را به شکلی سیستماتیک فرموله کرده و آنرا به همراه سایر مدلهای ریاضی در سیستمهای مهندسی قرار دهد.

تئوری فازی در مقاله ای به نام "مجموعه فازی" معرفی گردید. zadeh در سال 1962 چیزی را بدين مضمون برای سیستم های بیولوژیک نوشت: "ما اساساً به نوع جدیدی از ریاضیات نیازمندیم، ریاضیات مقادیر مبهم یا فازی که توسط توزیع های احتمالات قابل توصیف نیستند." پس از آن، وی ایده اش را در مقاله "مجموعه فازی" تجسم بخشید. با پیدایش تئوری فازی بحث و جدل ها پیرامون آن آغاز گردید. بعضی ها آنرا تأیید کرده و کار روی این زمینه حدید را آغاز کردند و برخی دیگر نیز این ایجاد را وارد می کردند که این ایده برخلاف اصول علمی موجود می باشد. با این حال بزرگترین چالش از ناحیه ریاضیدانانی بود که معتقد بودند تئوری احتمالات برای حل مسائلی که تئوری فازی ادعای حل بهتر آن را دارد، کفایت می کند. بدلیل اینکه

کاربردهای عملی تئوری فازی در ابتدای پیدایش آن مشخص نبود ، تفهیم آن از جهت فلسفی کار مشکلی بود و تقریباً هیچیک از مراکز تحقیقاتی تئوری فازی را به عنوان یک زمینه تحقیق جدی نگرفتند.

با وجودی که تئوری فازی جایگاه واقعی خود را پیدا نکرد ، با این حال هنوز محققینی بودند که در گوشه و کنار دنیا ، خود را وقف این زمینه جدید نمودند و در اواخر دهه 1960 روش‌های جدید فازی نظری الگوریتم‌های فازی ، تصمیم‌گیری‌های فازی و ... را مطرح کردند.

اگر بگوئیم پذیرفته شدن تئوری فازی بعنوان یک زمینه مستقل بواسطه کارهای بر جسته پروفسور لطفی زاده بوده ، سخن به گراف نگفته ایم. بسیاری از مفاهیم بنیادی تئوری فازی بوسیله ایشان در اواخر دهه 60 و اوایل دهه 70 مطرح گردید. پس از معرفی مجموعه‌های فازی در سال 1965 ، او مفاهیم الگوریتم‌های فازی در سال 1968 ، تصمیم‌گیری فازی در سال 1970 و ترتیب فازی را در سال 1971 مطرح نمود.

در مجموع ، پایه گذاری تئوری فازی در دهه 1970 صورت گرفت. با معرفی مفاهیم جدید ، تصویر تئوری فازی بعنوان یک زمینه جدید ، هر چه بیشتر شفاف گردید. کاربردهای اولیه ای نظری کنترل موتور بخار و کنترل کوره سیمان نیز تئوری فازی را به عنوان یک زمینه جدید مطرح کرد. معمولاً زمینه‌های تحقیق جدید باید بوسیله مراکز تحقیقاتی و دانشگاهها حمایت گردد. این امر متأسفانه در مورد تئوری فازی اتفاق نیفتاد. ضمن اینکه بسیاری از محققین ، تحقیقات خود را در مورد تئوری فازی بدليل عدم پشتیبانی تغییر دادند. این موارد بیشتر در ایالات متحده اتفاق افتاد.

در سال 1980 سوگنو شروع به ساخت اولین کاربرد ژاپنی فازی ، یعنی کنترل تصفیه آب فوجی نمود، و در سال 1983 مشغول کار بر روی یک رباط فازی شد. ماشینی که از راه دور کنترلر می شد و خودش به تنهایی عمل پارک را انجام می داد. در همین سالها یا شانوبو و میاموتو از شرکت هیتاچی کار بر روی سیستم کنترل قطار زیرزمینی سندایی را آغاز کردند. که این پروژه در سال 1987 به ثمر رسید که یکی از پیشرفت‌های ترین سیستم‌های قطار زیرزمینی در جهان بود. در جولای 1987 ، دومین کنفرانس سیستم‌های فازی در توکیو برگزار گردید. این کنفرانس درست سه روز بعد از افتتاح قطار زیرزمینی سندایی آغاز به کار کرد. در این کنفرانس هیروتا رباتی را به نمایش گذاشت که پینگ‌پنگ بازی می کرد و یاماکاوا سیستمی را به نمایش گذاشت که یک پاندول معکوس را در حالت تعادل قرار می داد و در اوایل دهه 90 تعداد زیادی از لوازم و وسائلی که بر اساس تئوری فازی کار می کردند ساخته شدند.

در فوریه 1992 اولین کنفرانس بین المللی IEEE در زمینه سیستمهای فازی در سان دیگو برگزار شد. این یک اقدام سمبیلیک در مورد پذیرش سیستمهای فازی بوسیله بزرگترین سازمان مهندسی یعنی IEEE بود. در سال 1993 بخش سیستمهای فازی IEEE گشایش یافت . از نقطه نظر تئوری سیستم‌های فازی و کنترل در اواخر دهه 80 و اوائل دهه 90 رشد چشمگیری پیدا کرد و پیشرفت‌هایی در زمینه برخی مشکلات اساسی سیستمهای فازی صورت گرفت. بعنوان مثال تکنیک‌های شبکه عصبی برای تعیین و تنظیم توابع تعلق استفاده شدند. با وجودی که تصویر سیستمهای فازی شفاف تر شده، با این حال کارهای زیادی باید هنوز باید انجام شود و بسیاری از راه حل‌ها و روش‌ها در ابتدای راه قرار دارد. ما اعتقاد داریم که تنها سرمایه گذاری مراکز تحقیقاتی معتبر بر روی افراد مستعد و خلاق می تواند باعث پیشرفت‌های عمدی و چشمگیری در زمینه تئوری فازی گردد.

## نظریه امکان

فرض کنید  $y$  یک متغیر باشد که می تواند مقادیری را از مجموعه  $X$  اتخاذ کند. توزیع امکان برای  $y$  را می توان به این گونه در نظر گرفت که بر روی مقادیری که می تواند به  $y$  منتنسب گردد، یک محدودیت فازی قرار داده می شود. بامثالی مفهوم محدودیت فازی را روشن می کنیم: عبارت فازی «علی جوان است» را در نظر بگیرید. عبارت رابه صورت نمادین نمایش می دهیم :

جوان = سن(علی) =  $\mu$

براساس مجموعه جهانی (0,120)=U که برای مجموعه فازی جوان در نظر میگیریم، یک تابع عضویت مثل  $\lambda$  برای آن تعریف می شود. موضوع محدودیت فازی آن است که اگرچه سن (علی) میتواند در محدوده U مقدار بگیرد، ولی با بیان عبارت جوان که هیچ گونه اطلاعات عددی در آن نیست، این محدودیت را به وجود آورده ایم که نمی توان به سن علی هر مقداری را انتساب داد، بلکه باید این مقدار در محدوده ای باشد که سن یک فرد جوان در آن محدوده است. از آن جایی که محدوده سن جوانی، صریحاً ودقیقاً مشخص نیست، این محدوده با استفاده از یک مجموعه فازی توصیف می شود (به شکل بالا توجه کنید) بنابر این توزیع امکان یک زیر مجموعه فازی روی مجموعه جهانی X تعریف می کند که تابع عضویت عناصر این زیر مجموعه فازی میزان امکان آن که این متغیر را مقداری از X بگیرد، مشخص می کند.

## تفاوت میان نظریه احتمالات و منطق فازی

مهمترین نکته ای که بر روی آن تکیه وپاشاری وجود دارد آن است که موضوع امکان وموضوع احتمال کاملاً متفاوت است، هرچند شباهت هایی باهم دارند و یکی از مباحث مهم در منطق فازی، تمیزدادن آن از نظریه احتمالات در علم ریاضیات است. غالباً نظریه فازی با نظریه احتمالات اشتباه می شود. در حالی که این دو مفهوم کاملاً با یکدیگر متفاوتند. این موضوع به قدری مهم است که حتی برخی از دانشمندان بزرگ علم ریاضیات در دنیا - بهویژه کشورهای غربی - درمورد آن با یکدیگر بحث دارند و جالب آن که هنوز هم ریاضیدانانی وجود دارند که با منطق فازی مخالفند و آن را یک سوء تعبیر از نظریه احتمالات تفسیر می کنند. بعضی از منتقدان در ابتدا معتقد بودند که نظریه فازی جز همان مسایل مطرح در بحث احتمال و آمارنیست و فقط بایان وظاهر دیگر معرفی می شود، ولیکن توسعه دهندها این نظریه با طرح مسایل ومفاهیمی ثابت کردند که این دوموضوع باهم ارتباطی ندارند. جالب این جاست که در عالم مجموعه های فازی بحث «اتفاقات فازی» و «احتمال و آمارفازی» به طور مجزا وجود دارد واین که نظریه فازی را مبتنی بر احتمال بدانیم اشتباه است.

آنچه در مورد نظریه فازی اهمیت دارد آن است که اطلاعاتی که به آن استناد می شود، کاملاً حقیقی ولیکن توأم با عدم دقت و صراحت هستند) همانند آنچه در دنیای واقعی اتفاق می افتد) وکاملاً با موضوع احتمال که مبنی بر پدیده های تصادفی است، فرق می کند. تنها نقطه اشتراک «نظریه احتمال» و «نظریه امکان» آن است که «عدم امکان»، «عدم احتمال» را نتیجه می دهد، ولی عکس آن صادق نیست .

از نگاه این ریاضیدانان، منطق فازی چیزی نیست جز یک برداشت نادرست از نظریه احتمالات که به گونه ای غیرقابل قبول، مقادیر و اندازه گیری های نادقيق را وارد علوم ریاضیات، مهندسی و کنترل کرده است. بعضی نیز مانند Bruno de Finetti معتقدند فقط یک نوع توصیف از مفهوم عدم قطعیت در علم ریاضیات کافی است و چون علم آمار و احتمالات وجود دارد، نیازی به مراجعة به منطق فازی نیست.

با این حال، اکثریت طرفداران نظریه منطق فازی، کارشناسان و متخصصانی هستند که به طور مستقیم یا غیرمستقیم با علم مهندسی کنترل سروکار دارند. حتی تعدادی از پیروان منطق فازی همچون بارت کاسکو تا آنجا پیش می روند که احتمالات را شاخه و زیرمجموعه ای از منطق فازی می نامند.

توضیح تفاوت میان این دو نظریه البته کار چندان دشواری نیست. منطق فازی با حقایق نادقيق سروکار دارد و به حدود و درجات یک واقعیت اشاره دارد؛ حال آن که نظریه احتمالات بر شالوده مجموعه حالات تصادفی یک پدیده استوار است و درباره شانس وقوع یک حالت خاص صحبت می کند؛ حالتی که وقتی اتفاق بیفتد، دقیق فرض می شود. ذکر یک مثال می تواند موضوع را روشن کند. فرض کنید در حال رانندگی در یک خیابان هستید. اتفاقاً متوجه می شوید که کودکی در اتومبیل دیگری که به موازات شما در حال حرکت است، نشسته و سر و یک دست خود را از پنجره ماشین بیرون آورده و در حال بازی گوشی است. این وضعیت واقعی است و نمی توان گفت احتمال این که بدن این کودک بیرون اتومبیل باشد، چقدر است.

چون بدن او واقعاً بیرون ماشین است، با این توضیح که بدن او کاملاً بیرون نیست، بلکه فقط بخشی از بدن او در خارج اتومبیل قرار گرفته است. تئوری احتمالات در اینجا کاربردی ندارد. چون ما نمی توانیم از احتمال خارج بودن بدن کودک از ماشین صحبت کنیم؛ زیرا آشکارا فرض غلطی است. اما می توانیم از احتمال وقوع حادثه صحبت کنیم. مثلاً هرچه بدن کودک بیشتر بیرون باشد،

احتمال این که در اثر برخورد با بدنه یک اتومبیل در حال حرکت دچار آسیب شود، بیشتر می‌شود. این حادثه هنوز اتفاق نیفتاده است، ولی می‌توانیم از احتمال وقوع آن صحبت کنیم. اما بیرون بودن تن کودک از ماشین همین حالا به واقعیت تبدیل شده است و فقط می‌توانیم از میزان و درجات آن صحبت کنیم.

تفاوت ظریف و در عین حال پرنگی میان نظریه احتمالات و نظریه فازی وجود دارد که اگر دقت نکنیم، دچار اشتباه می‌شویم؛ زیرا این دو نظریه معمولاً در کنار یکدیگر و در مورد اشیای مختلف همزمان مصداق‌هایی پیدا می‌کنند. هنگامی که به یک پدیده می‌نگریم، نوع نگاه ما به آن پدیده می‌تواند تعیین کند که باید درباره احتمالات صحبت کنیم یا منطق فازی. در مثال فوق موضوع دغدغه ما کودکی است که در حال بازی گوشی است. اما یک وقت نگران این هستیم که تا چه اندازه خطر او را تهدید می‌کند. خطروی که هنوز به وقوع نپیوسته است. یک وقت هم ممکن است نگران باشیم که بدن او چقدر بیرون پنجره است. واقعیتی که هم‌اکنون به وجود پیوسته است.

یک دیدگاه درباره علت بحث و جدل علمی میان دانشمندان این است که برخی از یاضیدانان اتکا به علم آمار و احتمال را کافی می‌دانند و نظریه فازی را یک برداشت غیرکارآمد از جهان درباره ما تلقی می‌کنند. به عنوان مثال، اگر به مورد کودک و اتومبیل مراجعه کنیم، این پرسش مطرح می‌شود که اگر نگرانی و دغدغه نهایی ما احتمال وقوع حادثه است، دیگر چه نیازی به این است که ما درباره درجات <بیرون بودن تن کودک از اتومبیل> صحبت کنیم؟

در اینجا برای نمایاندن تفاوت موضوع امکان و موضوع احتمال مثال جالب و معروفی وجود دارد که مطرح می‌کنیم:  
فرض، کنید یک نفر در بیابانی خشک وبی آب و علف نیاز مبرمی به آب دارد. دو بطری حاوی مایع کند که روی اولی نوشته شده است که این بطری به احتمال ۹۵٪ محتوی آب و با احتمال ۵٪ حاوی مواد غیر آشامیدنی است. بر روی بطری دوم این عبارت درج شده که مایع موجود در این بطری با درجه ۹۵٪ به مجموعه آب‌های آشامیدنی و بادرجه عضویت ۵٪ به مجموعه آب‌های غیر آشامیدنی تعلق دارد. یک عقل سلیم در چنین شرایطی کدام بطری را انتخاب می‌کند؟

اگر شخصی از بطری اول استفاده کند، با احتمال ۹۵ درصد به آب سالم دسترسی پیدا کرده است و بدون هیچ مشکلی رفع عطش می‌کند، ولی با احتمال ۵ درصد ممکن است بطری حاوی مواد غیر آشامیدنی بوده و شخص کاملاً مصدوم شود. حال اگر شخص از بطری دوم استفاده کند، از آن جایی که محتوای آن بادرجه عضویت ۹۵ درصد به آب‌های آشامیدنی تعلق دارد، بنابر این شخص مطمئن است که اگرچه آب صد درصد سالمی نمی‌نوشد، ولی تا حدود ۹۵ درصد به آب شبیه است و برای نوشیدن آن نخواهد مرد. بنابراین عقل حکم می‌کند باید راهی را بر گزید که در آن مخاطره‌ای وجود نداشته باشد.  
 البته بحث درباره ابعاد فلسفی منطق فازی بسیار شیرین و البته گسترده است. متأسفانه مجال برای طرح گسترده ابعاد فلسفی منطق فازی در اینجا وجود ندارد.

## منطق فازی :

در علوم فنی و مهندسی و علوم پایه، دانشمندان برای مدلسازی پدیده‌ها یک رابطه‌ی به ظاهر دقیق ریاضی ارایه کرده‌اند (هرچند در کار برد های عملی از دقت‌های آنچنان بالا استفاده نمی‌کنند ولی در دنیای پیرامون ما، هر آنچه اتفاق می‌افتد، مدل ریاضی ساده و ثابتی ندارد. در مسایلی که در عمرمان تجربه کرده‌ایم، مواردی وجود دارد که هیچ گاه نتوانسته ایم ویا حتی نخواسته ایم معیار کمی عددی روی آن بگذاریم. به عنوان مثال در محاورات روز مره از عباراتی مثل «نzdیک» یا «دور» برای بیان فاصله یا «گران» «ازان» برای ارزیابی قیمت استفاده می‌کنیم).

اگرچه عبارت «نzdیک» اطلاعات دقیقی در خود ندارد، ولی یک احساس ذهنی از میزان فاصله به شنونده ارایه کرده است و ذهن بر مبنای آن می‌تواند تصمیم گیری کند (این تصمیم گیری به هیچ وجه بر مبنای احتمال نیست). در دنیای واقعی، کنترل و نظارت

بسیاری از پدیده ها و فرآیندها به دقت نیاز دارند. حتی بسیاری از این پدیده ها و فرآینده ها با مدل های ریاضی قابل بیان و توجیه نیستند. شاید نتوان عملکرد ماشینی را که با توجه به چند قاعده حسی و ذهنی بسیار ساده و با یک دقت کاملاً معمولی توسط انسان کنترل می شود، با فرمول و قواعد ریاضی و محاسباتی مدل سازی و کنترل کرد.

منطق فازی (fuzzy)، مفهوم مجموعه ها، کمیت های نا دقیق و همراه با عدم صراحة رادریک روال کاملاً منظم و قانونمند معرفی کرده است و روش های تصمیم گیری در چنین محیط هایی را پیشنهاد می کند. از آن جایی که منطق فازی با واقعیت ها و حقایقی که فکر بشر با آن هادرگیر است، بیشتر منطبق است و از طرفی مدل سازی پدیده های فازی را با استفاده از نظریه فازی ساده تر و حقیقی تر به نظر می رسد، این نظریه در بسیاری از علوم و گرایش های گوناگون مورد استفاده قرار گرفته تا جایی که امروزه، گذشته از ابداع و تکمیل کامپیوترهای مبتنی بر منطق فازی، این منطق در تجزیه و تحلیل و تصمیم گیری در مورد مسائل سیاسی و اقتصادی نیز وارد شده است.

اگر چه واژه منطق فازی تداعی کننده منطق بی اعتبار است، ولی دقیقاً به منطق تخمینی و تقریبی ارجاع داده می شود. برخلاف منطق بولی، که به هر گزاره ارزش درست یا غلط را نسبت می دهد، منطق فازی درجه درستی به گزاره ها نسبت می دهد. کامپیوتر ها می توانند از منطق فازی برای نشان دادن ایده های مبهم و غیر دقیق استفاده کنند. مانند گرما و بلندی. در اینجا به مفاهیم بنیادی منطق فازی خواهیم پرداخت.

#### مفاهیم اساسی:

سیستم منطق کلاسیک بر پایه منطق بولی است، منطق بولی بر این فرض استوار است که یک عنصر یا عضو مجموعه داده شده است و یا عضو مجموعه مفروض نیست. هر دو فرض فوق نمی توانند تواما درست باشند. متأسفانه این سیستم برای نشان دادن مفاهیم مبهم محدودیت دارد. به عنوان مثال، فرض کنید منطق بولی برای تشخیص اینکه یک اتاق گرم است یا سرد مورد استفاده قرار گیرد. همه افراد با این فرض که 100 درجه فارنهایت برای دمای اتاق گرم و 25 درجه فارنهایت برای دمای اتاق سرد محسوب می شود موافقند. اما اگر دمای اتاق 75 درجه فارنهایت باشد، دسته بندی دمای اتاق بر حسب سرد و یا گرم بودن اتاق بسیار مشکل خواهد بود. در این مورد و موارد مشابه منطق بولی وسیله‌ی مناسبی برای شناسایی مقدار میانه نیست. منطق فازی روش توسعه یافته منطق بولی برای به کاربردن مفاهیم مبهم است. برای بیان ابهام در قالب یک عدد، منطق فازی تابعی برای عضویت در یک دسته معرفی می کند، که به هر عنصر یک عدد حقیقی بین صفر و یک نسبت می دهد (صفر و یک هم شامل این اعداد می باشند). این عدد نشان دهنده درجه عضویت عنصر نسبت به مجموعه مورد نظر می باشد. عضویت صفر بیانگر این است که عنصر مورد نظر کاملاً خارج از مجموعه است. در حالی که عدد یک نشان دهنده این است که عنصر مورد نظر کاملاً در مجموعه قرار دارد. هر عدد بین این دو مقدار بیانگر درجه عضویت بخشی می باشد. حال بیایید مثالی را که در پاراگراف قبل بررسی شد مورد بحث قرار دهیم. اگر منطق فازی برای بررسی گرمی اتاق مورد استفاده قرار گیرد، 100 درجه فارنهایت، دارای مقدار یک و 25 درجه فارنهایت مقدار صفر را دارا خواهد بود. از طرف دیگر 75 درجه فارنهایت مقداری بین صفر و یک خواهد داشت.

#### توضیح اضافی در مورد تابع عضویت:

یک متخصص دانا باید تابع عضویتی ارائه دهد که با عقاید عمومی سازگاری داشته باشد. تابع عضویتی که گرمی یک اتاق را توصیف می کند، بایستی مفهوم سردی و گرمی را که در ذهن افراد است منعکس نماید. اگرچه منطق فازی بر تابع عضویت تکیه دارد، ولی سرچشمۀ آن خارج قلمرو این تابع می باشد. این تابع می تواند اشکال مختلفی داشته باشد: مثلثی، زنگی، ذوزنقه‌ای، نمایی و... شکل زیر نشان دهنده چندین تابع برای نشان دادن گرمی اتاق است. در این توابع دمای 75 درجه فارنهایت به عنوان دمای میانگین در نظر گرفته شده است.

#### نمایش گرافیکی سه تابع عضویت مختلف برای نشان دادن گرمی اتاق:

از آنجا که منطق فازی توسعه یافته منطق بولی است، مفاهیم منطقی مشترک زیادی می‌توانند در هر دو اعمال شوند. بیویژه اعمال کلاسیکی مانند: اجتماع، اشتراک و متمم در منطق فازی تعریف می‌شوند. در عین حال این مفاهیم دارای تعاریف متفاوتی هستند.

$$\begin{aligned} \text{NOT } x &= (1 - u(x)) \\ x \text{ AND } y &= \min(u(x), u(y)) \\ x \text{ OR } y &= \max(u(x), u(y)) \end{aligned}$$

مانند منطق بولی، منطق فازی نیز می‌تواند از قانون «اگر <شرط> آنگاه <عمل>» استفاده کند. برای مثال قانونی برای تهویه مطبوع می‌تواند به این صورت باشد: «اگر اتاق گرم و مرطوب است آنگاه دستگاه را روشن کن» اما برخلاف منطق بولی، قسمت شرط با عبارات صحیح یا غلط سنجیده نمی‌شود، بلکه با درجه درستی مورد ارزیابی قرار می‌گیرد.

با این تعاریف منطق فازی دارای این قدرت است که در تنظیم سیستم‌ها از میکرو کنترلرهای ساده و کوچک و جاسازی شده گرفته تا PC‌های چند کاناله شبکه شده بزرگ یا سیستم‌های کنترلی به کار برده شود. این منطق دارای قدرت اجرایی در سخت افزار، نرم افزار یا ترکیبی از هر دوی اینهاست. در واقع منطق فازی راه ساده‌ای را برای رسیدن به یک نتیجه قطعی و معین بر پایه اطلاعات ورودی ناقص، خطای دار، مبهم و دوپهلو فراهم می‌کند. منطق فازی یک قانون ساده بر مبنای

" IF x And y THEN z"

را بیان می‌کند.

به عنوان مثال به جای برخورد با اصطلاحاتی نظیر "T<1000F", "210<TEMP<220", "SP=500F" ، اصطلاحاتی نظیر

"IF (process is too cool) AND (process is getting colder) THEN

(Add heat to the process) "

Or

"IF (process is too hot) AND (process is heating rapidly) THEN

(Cool the process quickly)"

به کار برده شود.

درست مثل کاری که در هنگام دوش گرفتن انجام می‌دهیم: در صورتی که آب خیلی سرد یا خیلی گرم باشد بدون اینکه از درجه دقیق آب اطلاعی داشته باشیم تنها بر اساس پردازش انجام شده در معزز به کمک دریافت دمای هوا از طریق حسگرهای پوست با کمی سختی کشیدن آب را به سرعت به دمای دلخواه در می‌آوریم یا آنکه میتوانیم در یک اتاق به اشیاء گوناگونی نگاه کنیم و تصمیم بگیریم کدامیک بیشتر شبیه صندلی است ویا به مردم نگاه کنیم و بگوییم کدامیک شبیه John Wayne ویا کدامیک بیشتر شبیه گاندی است. منطق فازی قادر به تقلید اینگونه رفتارها اما با سرعت بسیار بالایی است. از طرفی باید به این نکته هم توجه کنیم که تمامی سیستم‌های طبقه بندی ساخته ذهن انسان هستند و برچسب درست تا زمانی به یک سیستم

طبقه بندی نسبت داده میشود که سیستم کنترلی دیگر آن را رد نکند مثلا در تئوری نسبیت دیگر درست نیست بگوییم زمین دور خورشید میگردد پس خورشید هم دور زمین می گردد! یا به عنوان مثال دیگر، کشف موجودی عجیب در استرالیا که پلاتی پوس نامیده می شودو برخلاف پستانداران دیگر همانند خزندگان تخم میگذارد و جوجه های جوان را شیر می دهد! با این تعاریف می توان گفت که منطق فازی یک تکنولوژی کنترلی بسیار قدرمند است که به جای ساختن یک حصار در اطراف یک طبقه بندی سعی دارد آن را به گونه ای توصیف کند که به ایده نزدیک تر است.

### متغیر های زبان شناختی :

در زندگی روزمره، کلماتی را به کار می بریم که اغلب برای توصیف متغیرها استفاده می شوند. به عنوان مثال هنگامیکه می گوییم "امروز سرد است" یا "دمای هوا امروز پایین است" از واژه "پایین" برای توصیف "دمای هوا امروز" استفاده کرده ایم به این معنی که متغیر دمای هوا امروز واژه "پایین" را به عنوان مقدار خود پذیرفته است. واضح است که متغیر "دمای هوا امروز" میتواند مقادیری نظری  $3^{\circ}, 8^{\circ}, 24^{\circ}$  را اختیار کند. هنگامیکه یک متغیر، اعداد را به عنوان مقدار پذیرد ما یک چهارچوب ریاضی مشخص برای فرموله کردن آن داریم اما هنگامیکه متغیر واژه ها را به عنوان مقدار میگیرد در آن صورت چهارچوب مشخص برای فرموله کردن آن در تئوری ریاضیات کلاسیک نداریم. در واقع در سیستم های عملی اطلاعات مهم از دو منبع سرچشمه می گیرند: یکی از منابع افراد خبره که دانش و آگاهیشان را در در مورد سیستم با زبان طبیعی تعریف میکنند و منبع دیگر اندازه گیری ها و مدل های ریاضی هستند که از قواعد فیزیکی مشتق شده اند. بنابر این یک مساله مهم ترکیب این دو نوع اطلاعات در طراحی سیستم هاست. برای انجام این ترکیب سوال کلیدی این است که چگونه می توانیم دانش بشری را به یک فرمول ریاضی تبدیل کنیم؟

برای اینکه چنین چهارچوبی به دست آوریم مفهوم متغیر های زبانی تعریف شده است. در صحبت های عامیانه اگر یک متغیر بتواند واژه هایی از زبان طبیعی را به عنوان مقدار بپذیرد یک متغیر زبان شناختی نامیده میشود. برای فرموله کردن واژه ها در گزاره های ریاضی از مجموعه های فازی برای مشخص کردن واژه ها استفاده میکنیم و تعریف میکنیم: "اگر یک متغیر بتواند واژه هایی از زبان طبیعی را به عنوان مقدار خود بپذیرد آنگاه متغیر زبان شناختی نامیده میشود که واژه ها بوسیله مجموعه های فازی در محدوده ای که متغیرها تعریف شده اند مشخص می گردد.

### مجموعه های فازی :

در نظریه مجموعه های فازی، اصول و قوانینی تعریف شده است که مجموعه های کلاسیک رانیز به عنوان یک حالت خاص از مجموعه های فازی در بر می گیرد. در مجموعه های که کلاسیک تابع عضویت فقط مقادیر صفر یا یک را شامل می شود (یعنی یک شی یا عضو یک مجموعه هست یا نیست) ولی در مجموعه های فازی این تابع به صورت درجه عضویت بیان می شود. هر شی در یک مجموعه فازی می تواند هم عضو یک مجموعه باشد وهم نباشد) اصولاً یک مجموعه فازی به صورت مجموعه ای از عناصر در «جهان اطلاعات» تعریف می شود، به صورتی که حدود مجموعه ای که در این مجموعه قرار دارد، مبهم یا به عبارتی فازی است. در مجموعه کلاسیک، تابع عضویت تابعی است به اصطلاح تیز و مشخص بادو مقدار صفر یا یک. به عبارت واضحتر، یک عضو  $X$  یا به مجموعه ای مثل  $A$  تعلق دارد یا ندارد و این دو حالت خارج نیست. ولی برای یک مجموعه فازی، تابع عضویت تابعی است با تغییر تدریجی نه آنی. یعنی مفهوم یا هست و یا نیست را کنار گذاشته واژ مفهوم میزان و درجه عضویت (چقدر هست، چقدر نیست) استفاده می شود. در حقیقت محدوده یک مجموعه فازی مبهم و مشکوک است و در یک کلام فازی است. برای روشن تر شدن موضوع فرض کنیم بخواهیم بین اعضای یک جامعه از نظر سنی مجموعه افراد نوجوان و مجموعه افراد جوان و مجموعه افراد مسن را تعریف کنیم. محدوده بین نوجوانی و جوانی و بیرونی را نمی توان به صورت دقیق مشخص کرد و هیچ محدوده صریح و مطمئنی برای ارزیابی وجود ندارد. نظریه مجموعه های فازی از همین مفهوم نشأت می گیرد. گذشته از معیار هایی که اصلانمی توان آن هارا اندازه گیری کرد (مثل زیبایی و زشتی، خوبی و بدی) بسیاری از کمیت ها در زندگی روزمره موجودند که برای ما مقدار آن ها اهمیتی ندارد و برای بیان

آن ها به طور ناخود آگاه از عبارات فازی استفاده می کنیم ( مثل سردی و گرمی هوا، نزدیکی یا دوری یک مقصد ). (جالب این جاست که همین عبارات فازی که هیچ گونه اطلاعات کمی دقیق در خود ندارند، به نحو مؤثری در تصمیم گیری های روزمره به کار می آیند .

اعضای یک مجموعه فازی دارای درجه عضویتی بین صفر و یک هستند. این مفهوم با نظریه مجموعه های کلاسیک متناقض است، زیرا در نظریه ی کلاسیک یک عنصر عضو یک مجموعه نیست، مگر آن که کاملاً در آن مجموعه وجودداشته باشد. با یک بیان حسی می توان گفت که یک عضو از یک مجموعه فازی با یک درجه عضویت می تواند به مجموعه های دیگر بادرجه عضویت متفاوت ( یا حتی مشابه ) متعلق باشد. مثلاً یک رنگ که بادرجه عضویت  $0/6$  متعلق به مجموعه فازی رنگ نارنجی است، می تواند بادرجه عضویت  $0/5$  به مجموعه فازی رنگ زرد و بادرجه عضویت  $0/3$  به مجموعه فازی رنگ قرمز تعلق داشته باشد. معمولاً به منظور تمایز مجموعه های فازی با یک علامت که در زیر یا بالای نام آن مشخص می شود. مثلاً :

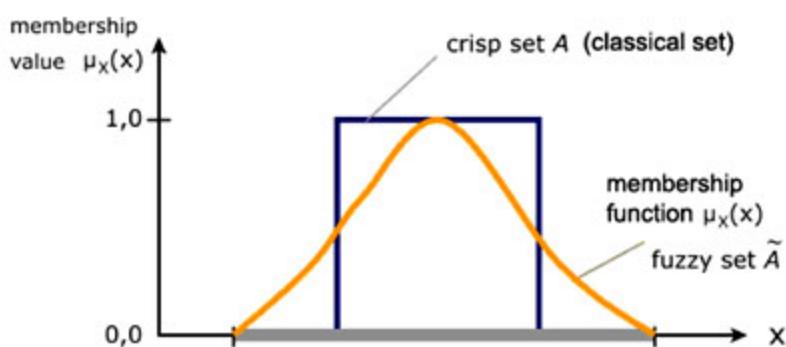
این جمله که <آقای الف به اندازه هفتاد درصد عضو جامعه بزرگسالان است> از دید تئوری مجموعه های فازی صحیح است. در این تئوری، عضویت اعضای مجموعه از طریق تابع  $(X) \rightarrow U$  مشخص می شود که  $X$  نمایانگر یک عضو مشخص و  $U$  تابعی فازی است که درجه عضویت  $X$  در مجموعه مربوطه را تعیین می کند و مقدار آن بین صفر و یک است ( فرمول زیر ).

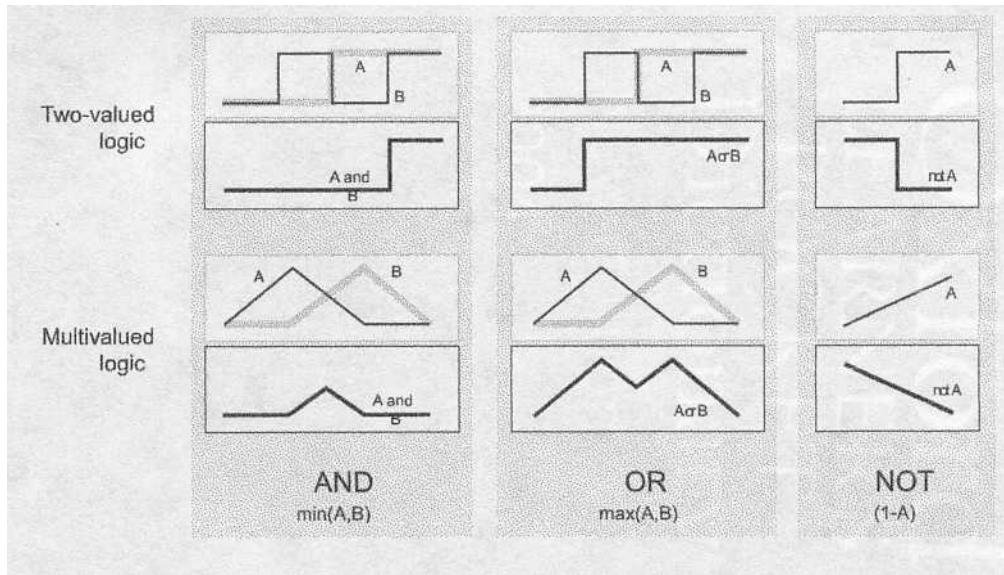
$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\}$$

فرمول 1

به بیان دیگر،  $(X) \rightarrow U$  نگاشتی از مقادیر  $X$  به مقادیر عددی ممکن بین صفر و یک را می سازد. تابع  $(X) \rightarrow U$  ممکن است مجموعه ای از مقادیر گسسته (discrete) یا پیوسته باشد. وقتی که  $U$  فقط تعدادی از مقادیر گسسته بین صفر و یک را تشکیل می دهد، مثلاً ممکن است شامل اعداد  $0/3$  و  $0/5$  و  $0/7$  و  $0/9$  و صفر و یک باشد. اما وقتی مجموعه مقادیر  $U$  پیوسته باشند، یک منحنی پیوسته از اعداد اعشاری بین صفر و یک تشکیل می شود.

شکل 1 نموداری از نگاشت پیوسته مقادیر  $X$  به مقادیر  $(X) \rightarrow U$  در این نمودار می تواند قانون عضویت در یک مجموعه فازی فرضی را تعریف کند.





### چند تعریف

**تعریف 2.1 :** یک مجموعه فازی  $A$  در فضای جهانی  $U$  بوسیله یک تابع  $\mu(x)$  که مقادیری در بازه  $[0, 1]$  اختیار می‌کند، مشخص می‌شود.

بنابراین یک مجموعه فازی تعمیم یک مجموعه کلاسیک است که اجزه می‌دهد تابع عضویت هر مقداری را در بازه  $[0, 1]$  اختیار کند. به عبارت دیگر یک مجموعه کلاسیک فقط می‌توانست دو مقدار  $0$  و  $1$  را اختیار کند. در حالیکه تابع عضویت یک مجموعه فازی، یک تابع پیوسته در محدوده  $[0, 1]$  می‌باشد. در واقع می‌بینیم که هیچ در مورد مجموعه فازی گنگ و مبهم نیست، بلکه، مجموعه با یک تابع عضویت پیوسته است.

یک مجموعه  $A$  را در  $U$  را می‌توان با یک مجموعه از زوجهای مرتب  $X$  و  $\mu(x)$  نمایش داد. بدین ترتیب:

$$A = \{(x, \mu(x)) | x \in U\}$$

تعریف: تکیه گاه (Support) مجموعه فازی  $A$  در فضای جهانی  $U$  یک مجموعه غیر فازی است که شامل تمامی عضوهایی که تابع عضویت آنها غیر صفر است، می‌باشد. یعنی:

$$\text{supp}(A) = \{x \in U | \mu(x) > 0\}$$

که  $(A)$  نشان دهنده تکیه گاه مجموعه فازی  $A$  است. اگر تکیه گاه یک مجموعه فازی هیچ عضوی نداشته باشد، آنرا یک مجموعه فازی تهی می‌نامند.

تعریف: یک منفرد فازی یک مجموعه فازی است که تکیه گاه آن یک مجموعه تک عضوی است.

تعریف: نقطه تقاطع یک مجموعه فازی نقطه‌ای است که تابع عضویت آن  $0.5$  است.

تعریف: ارتفاع یک مجموعه فازی، برابر با بزرگترین مقدار تابع عضویت آن مجموعه می‌باشد.

تعریف: یک مجموعه طبیعی فازی، مجموعه‌ای است که ارتفاع آن برابر با یک باشد.

## اعداد فازی :

یکی از بحث های جالب و کار بردی نظریه امکان در بحث منطق فازی، اعداد فازی هستند. این اعداد و خصوصیات آن ها توسط دوبو (Dubois) و پرید (prade) در سال های 1978 و 1979 و همچنین توسط szadeh در سال 1982 معرفی شدند.

یک عدد فازی در ساده ترین تعریف یک عدد حقیقی است که با یکتابع توزیع امکان محدود شده است، یعنی روی دامنه اعداد حقیقی که می توان به آن نسبت داد، محدودیت گذاشته ایم و این محدودیت یک زیر مجموعه فازی را مشخص می کند. مثال های بسیار ساده و روشنی از اعداد فازی عبارتند از:

خیلی نزدیک به 5 ، خیلی بزرگتر از 6، تقریباً 4

موضوعی که مایلم برای نتیجه گیری از مقدمات فازی در این نوشتار روی آن تأمل شود، آن است که برای کنترل و تصمیم گیری در دنیای واقعی از همین اعداد فازی زیاد استفاده می شود. البته باید توجه کرد که اعداد فازی بادقت های گوناگون قابل تعریف اند و براحتی می توان آن ها را همانند اعداد معمولی در محاسبات فازی دخالت داد. به عنوان مثال در محاورات روز مرہ، عبارت فازی زیر رابه دفعات شنیده ایم :

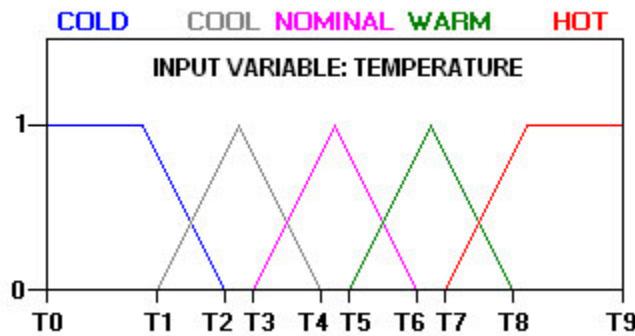
«قیمت جنس X تقریباً دوهزار افغانی است» و «جنس Y از حدود هشتتصد افغانی است» بنابر این باید برای خرید این دو قلم کالا تقریباً سه هزار افغانی» هزینه کرد. همان گونه که ملاحظه می شود عبارت تقریباً «سه هزار افغانی» که نتیجه محاسبات قیمت X و Y است، از نظر ذهن ما کاملاً منطقی است، هر چند به طور مستقیم از جمع دوهزار و هشتتصد به دست نیامده است .

## کنترل فازی

در کنترل فازی ، منطق فازی را از طریق قوانینی که <عملگرهای فازی> نامیده می شوند، می توان به کار گرفت. این قوانین معمولاً بر اساس مدل زیر تعریف می شوند:

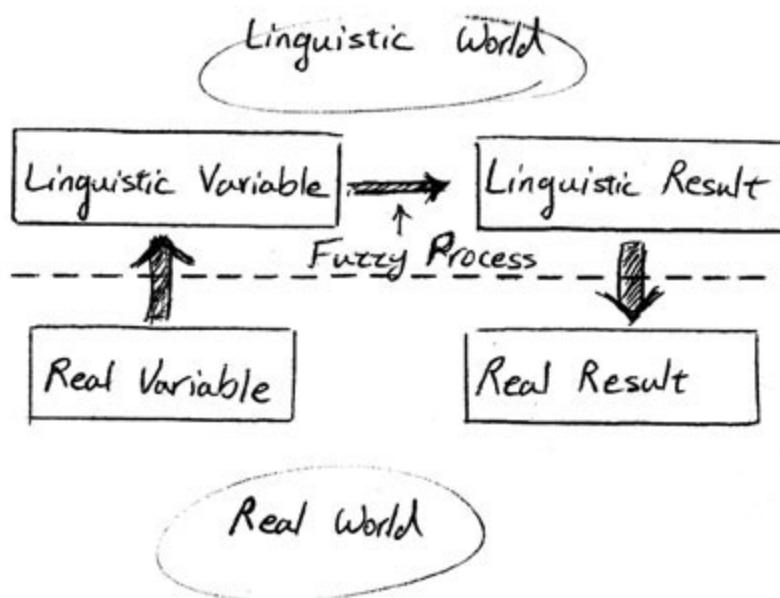
IF variable IS set THEN action

به عنوان مثال فرض کنید می خواهیم یک توصیف فازی از دمای یک اتاق ارائه دهیم. در این صورت می توانیم چند مجموعه فازی تعریف کنیم که از الگوی تابع ( $X \rightarrow$  تبعیت  $A$ ) نموداری از نگاشت متغیر <دمای هوای> به چند مجموعه فازی با نامهای <سرد>، <خنک>، <عادی>، <گرم> و <داغ> است. چنان که ملاحظه می کنید، یک درجه حرارت معین ممکن است متعلق به یک یا دو مجموعه باشد.



شکل 2

به عنوان نمونه، درجه حرارت‌های بین دمای  $T_1$  و  $T_2$  هم متعلق به مجموعه «سرد» و هم متعلق به مجموعه «خنک» است. اما درجه عضویت یک دمای معین در این فاصله، در هر یک از دو مجموعه متفاوت است. به طوری که دمای نزدیک  $T_2$  تنها به اندازه چند صدم در مجموعه «سرد» عضویت دارد، اما نزدیک نوددرصد در مجموعه «خنک» عضویت دارد.



پارادایم حاکم بر یک کنترل فازی به این ترتیب است که متغیرهای دنیای واقعی به عنوان ورودی دریافت می‌شوند. قوانین فازی آن‌ها را به متغیرهای معنایی تبدیل می‌کند. فرآیند فازی این ورودی را می‌گیرد و خروجی معنایی تولید می‌کند و سرانجام خروجی‌ها به زبان دنیای واقعی ترجمه می‌شوند. نمودار شکل 3 مصادقی از همین روند است.

اکنون می‌توان بر اساس مدل فوق قانون فازی زیر را تعریف کرد:

اگر دمای اتاق «خیلی گرم» است، سرعت پنکه را «خیلی زیاد» کن.

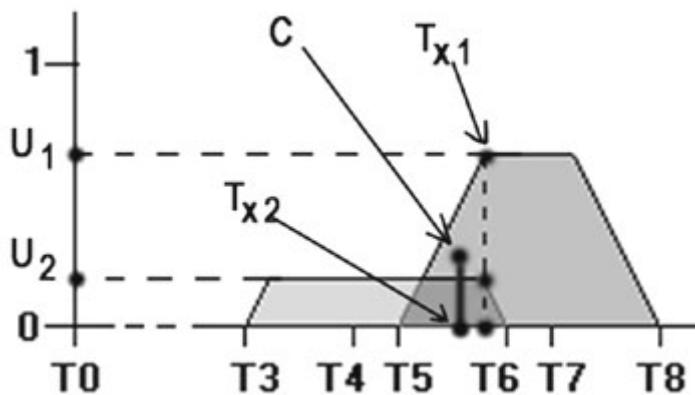
اگر دمای اتاق «گرم» است، سرعت پنکه را «زیاد» کن.

اگر دمای اتاق «معتدل» است، سرعت پنکه را در «همین اندازه» نگهدار.

اگر دمای اتاق «خنک» است، سرعت پنکه را «کم» کن.

اگر دمای اتاق <سرد> است، پنکه را <خاموش> کن.

اگر این قانون فازی را روی یک سیستم کنترل دما اعمال کنیم، آن‌گاه می‌توانیم دماسنجی بسازیم که دمای اتاق را به صورت خودکار و طبق قانون ما، کنترل می‌کند. اما این سؤال پیش می‌آید که اگر دو یا چند قانون همزمان برای یک متغیر ورودی فعال شود چه اتفاقی خواهد افتاد؟ فرض کنید دمای اتاق برابر  $T_x$  است در این صورت هم قانون مربوط به اتاق گرم و هم قانون مربوط به دمای اتاق معتمد صادق است و مقادیر  $U_1$  و  $U_2$  به ترتیب به دست می‌آید. طبق کدام قانون باید عمل کرد؟ لطفی‌زاده خود پاسخ این معما را نداد. در سال 1975 دو دانشمند منطق فازی به نام ممدانی (Mamdani) و آسیلیان اولین کنترل فازی واقعی را طراحی کردند. آنان پاسخ این معما را با محاسبه نقطه ثقل (C) مساحتی که از ترکیب دو ذوزنقه زیر  $U_1$  و  $U_2$  در شکل زیر پدید آمده و نگاشت آن به محور  $t$  و به دست آوردن مقدار  $T_x$  حل کردند.



اما منطق فازی برای گرفتن تصمیمات کاربردی در کنترل فازی چه می‌کند؟ روش چهار مرحله‌ای استفاده از منطق فازی در کنترل فازی عبارتست از:

**1) فازی کردن**

**2) استنتاج**

**3) ترکیب و ساخت**

**4) بر گرداندن از حالت فازی**

**1) فازی کردن:** در این مرحله واقعیات بر اساس سیستم فازی تعریف می‌شوند. ابتدا باید ورودی و خروجی سیستم معرفی شده، سپس قوانین اگر-آنگاه مناسب به کار گرفته شوند. برای ساختتابع عضویت بايستی از داده‌های خام استفاده شود. حال سیستم برای اعمال منطق فازی آماده است.

مثالی برای فازی کردن: دستگاه تهویه ای را در نظر بگیرید که با اندازه گیری دما و رطوبت اتاق میزان به جریان در آوردن هوای مشخص می‌کند. در این مورد ورودی عبارتست از دما و میزان رطوبت و خروجی سطح جریان هوای خروجی از دستگاه تهویه مطبوع است که شامل سه حالت، خاموش، کم و زیاد می‌باشد. همچنین این قوانین اگر-آنگاه استفاده می‌شوند:

**1) اگر اتاق گرم باشد آنگاه هوای زیادی منتشر کن**

**2) اگر اتاق خنک باشد آنگاه هیچ هوایی منتشر نکن**

**(3) اگر اتاق سرد و مرطوب است، آنگاه کمی هوا را به جریان بینداز.**

در آخر، یک متخصص دانا باید دوتابع عضویت تعیین کند. یکی برای اینکه دما را به مقدار فازی تبدیل کند و دیگری برای تبدیل میزان رطوبت به مقدار فازی

**(2) استنتاج: هنگامی که ورودی ها به سیستم می‌رسند استنتاج، همه قوانین اگر-آنگاه را مورد ارزیابی قرار می‌دهد و درجه درستی آنها را مشخص می‌کند. اگر یک ورودی داده شده به طور صریح با یک قانون اگر-آنگاه مشخص نشده باشد، آنگاه تطابق بخشی مورد استفاده قرار می‌گیرد تا جوابی مشخص شود. راههای متعددی برای پیدا کردن پاسخ بخشی وجود دارد که البته فراتر از حد این مقاله میباشند.**

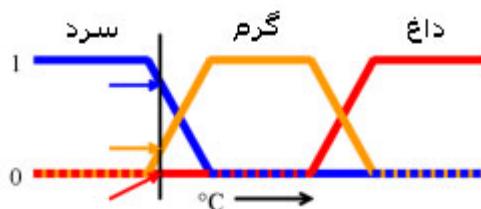
**مثالی برای استنتاج:** فرض کنید دستگاه تهویه مطبوع دما و درجه رطوبت را اندازه‌گیری کرده و به آنها به ترتیب مقادیر فازی 0.7 و 0.1 را نسبت داده باشد. حال این سیستم بایستی درستی هر یک از قوانین فازی را که در بالا بحث شد مورد بررسی قرار دهد. برای این منظور روش‌های استنتاج بسیاری وجود دارد. این مثال ساده ترین روش را مورد استفاده قرار می‌دهد که روش ماکسیمم-مینیمم نامیده می‌شود. این روش مقدار فازی قسمت آنگاه (نتیجه) را به قسمت اگر نسبت میدهد. بنابراین این روش مقادیر فازی 0.7 و 0.1 را به ترتیب به قوانین 1، 2 و 3 نسبت می‌دهد.

**(3) ساخت:** در این قسمت برای بدست آوردن یک نتیجه کلی تمامی مقادیر بدست آمده از قسمت استنتاج با هم ترکیب می‌شوند. قوانین فازی مختلف نتایج مختلفی خواهند داشت. بنابرایت ضروری است تا همه قوانین در نظر گرفته شوند. برای این منظور روش‌های متعددی وجود دارند که توضیح همه آنها در این مقاله نمی‌گنجد.

**مثالی برای ساخت:** هر نتیجه استنتاجی درباره سیستم تهویه مطبوع عمل خاصی را پیشنهاد می‌کند. در مثال فوق قانون اول، سطح گردش هوای زیاد را پیشنهاد می‌کند. قانون دوم، خاموش کردن، و قانون سوم، سطح گردش هوای کم را بیان می‌کند. تکنیک‌های متعددی برای بدست آوردن نتیجه کلی وجود دارند. این مثال از روش ماکزیمم-مینیمم که روش ساده‌ای است استفاده می‌کند. این روش ماکزیمم مقدار فازی قسمت استنتاج به عنوان نتیجه در نظر می‌گیرد. یعنی در عمل، قسمت ساخت مقدار 0.7 را انتخاب می‌کند چون مقدار بیشتری را بین مقادیر فازی دارا است.

**(4) بازگرداندن از حالت فازی:** در این مرحله مقدار فازی بدست آمده از قسمت ساخت به یک داده قابل استفاده تبدیل می‌شود. این قسمت از کار اغلب پیچیده است چون مجموعه فازی نبایستی مستقیماً به داده قابل استفاده تبدیل شود. از آنجا که کنترلگرهای سیستم‌های فیزیکی به سیگنال‌های گسسته نیاز دارند، این مرحله بسیار مهم می‌باشد.

**مثالی برای بازگرداندن از حالت فازی:** به خاطر دارید که مقدار فازی بدست آمده از مرحله قبل 0.7 بود. این مقدار عددی برای سیستم نهفته مطبوع قابل فهم نیست. باید مشخص شود که دستکاه کدامیک از فرامین کم، زیاد یا خاموش را به جریان بیندازد. مرحله بازگرداندن از حالت فازی بایستی عدد 0.7 را به یکی از فرامین فوق تبدیل کند. در این مثال واضح است که مقدار خروجی 0.7 بیانگر این است که سیستم تهویه مطبوع بایستی در حالت زیاد باشد.



## کاربردهای منطق فازی:

منطق فازی کاربردهای متعددی دارد. ساده‌ترین نمونه یک سیستم کنترل دما یا ترموموستات است که بر اساس قوانین فازی کار می‌کند. سال‌هاست که از منطق فازی برای کنترل دمای آب یا میزان کدرشدن آبی که لباس‌ها در آن شسته شده‌اند در ساختمان اغلب ماشین‌های لباسشویی استفاده می‌شود.

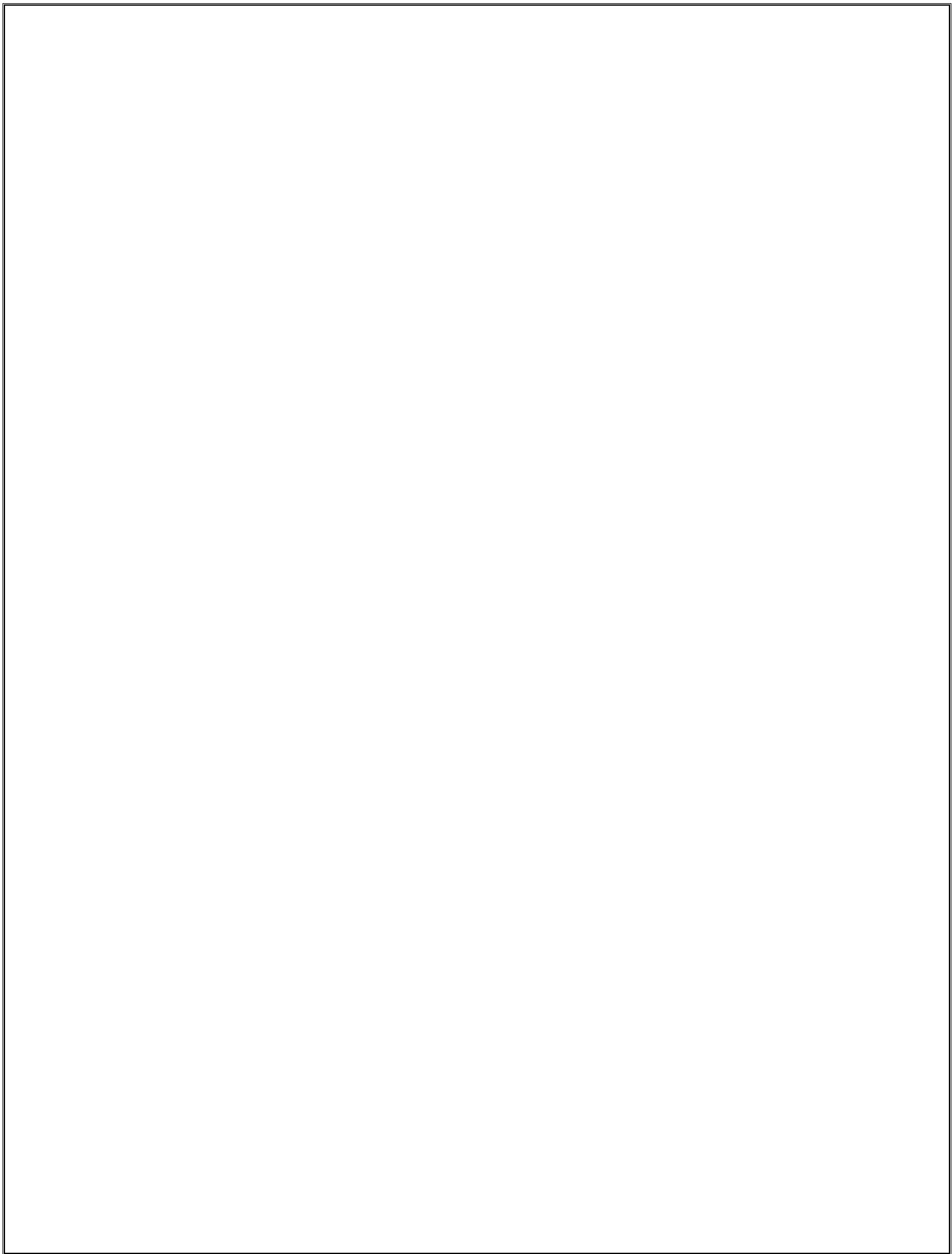
امروزه ماشین‌های ظرفشویی و بسیاری از دیگر لوازم خانگی نیز از این تکنیک استفاده می‌کنند. منطق فازی در صنعت خودروسازی نیز کاربردهای فروانی دارد. مثلاً سیستم ترمز و ABS در برخی از خودروها از منطق فازی استفاده می‌کند. یکی از معروف‌ترین نمونه‌های به کار گیری منطق فازی در سیستم‌های ترابری جهان، شبکه مونوریل (قطار تک ریل) توکیو در ژاپن است. سایر سیستم‌های حرکتی و جابه‌جایی بار، مثل آسانسورها نیز از منطق فازی استفاده می‌کنند.

سیستم‌های تهویه هوا نیز به وفور منطق فازی را به کار می‌گیرند. از منطق فازی در سیستم‌های پردازش تصویر نیز استفاده می‌شود. یک نمونه از این نوع کاربردها را می‌توانید در سیستم‌های < تشخیص لبه و مرز > اجسام و تصاویر<sup>(3)</sup> مشاهده کنید که در رباتیک نیز کاربردهایی دارد. به طور کلی خیلی از موقع در ساختمان سیستم‌های تشخیص الگوها (Pattern Recognition) مثل سیستم‌های تشخیص گفتار و پردازش تصویر از منطق فازی استفاده می‌شود.

## کاستی ها:

منطق فازی و منطق بولی هر دو بر پایه واقعیات می‌باشند. با این تفاوت که منطق فازی توانایی کارکردن با داده‌های مبهم را نیز دارد. با این وجود منطق فازی هنوز قادر به حل بعضی مسائل نیست: عضویت در یک مجموعه فازی شدیداً بر پایه داده‌های معین است. به عبارت دیگر، منطق فازی هیچ ادراکی از گمان‌ها، تعقل، شک یا ناسازگاری شواهد ندارد. بسیاری از سیستم‌ها، مانند آنچه در بحث کاربرد گفته شدمی توانند از منطق فازی بدون هیچ مشکلی استفاده کنند. چون نیاز به هیچ تصمیم گیری درونی و فکری ندارند. اما بعضی سیستم‌ها به منطق پیچیده تری نیاز دارند تا بتوانند به بیان گمان، تعقل و ... بپردازنند.

برای نشان دادن نقصان منطق فازی می‌توان به سیستم شناخت دوست یا دشمن اشاره کرد (**or Identify Friend or Foe : IFF**) این سیستم برای شناخت هوایپیماهای ارتشی یا مسافربری دوست و دشمن به کار می‌رود. در حالت عادی سیستم از هر هوایپیما یک سیگنال شناسایی دریافت می‌کند. دلایل زیادی وجود دارند که ممکن است سیستم این سیگنال را دریافت نکند مانند: بد عمل کردن سیستم، بد عمل کردن فرستنده، نبودن این سیستم روی هوایپیماها، پارازیت سیگنال یا خاموش بودن رادیو. در این موارد این سیستم باید از منطق برای شناسایی هوایپیماها استفاده کند. که فقط مربوط به داده‌های مبهم از قبیل مسیر پرواز نیست. چون این داده‌ها هوایپیماهای دوست را از دشمن تشخیص نمی‌دهند. بنابراین منطق فازی در سیستم تشخیص دوست از دشمن کاربردی ندارد. برای این منظور باید از منطقی استفاده شود که توانایی تصمیم گیری درونی را داشته باشد.



فصل دوم

# پردازش تصویر

## تصویر دیجیتالی چیست؟

یک تصویر را می‌توان توسطتابع دوبعدی  $f(x,y)$  که در آن  $X$  و  $Y$  را مختصات مکانی و مقدار  $f$  در هر نقطه را شدت روشنایی تصویر در آن نقطه می‌نامند. اصطلاح سطح خاکستری نیز به شدت روشنایی تصاویر مونوکروم اطلاق می‌شود. تصاویر رنگی نیز از تعدادی تصویر دوبعدی تشکیل می‌شود.

زمانی که مقادیر  $X$  و  $Y$  و مقدار  $f(x,y)$  با مقادیر گسسته و محدود بیان شوند، تصویر را یک تصویر دیجیتالی می‌نامند. دیجیتال کردن مقادیر  $X$  و  $Y$  را Sampling و دیجیتال کردن مقدار  $f(x,y)$  را quantization گویند.

برای نمایش یک تصویر  $N * M$  از یک آرایه دو بعدی (ماتریس) که  $M$  سطر و  $N$  ستون دارد استفاده می‌کنیم. مقدار هر عنصر از آرایه نشان دهنده شدت روشنایی تصویر در آن نقطه است. در تمام توابعی که پیاده سازی خواهیم کرد، هر عنصر آرایه یک مقدار 8 بیتی است که می‌تواند مقداری بین 0 و 255 داشته باشد. مقدار صفر نشان دهنده رنگ تیره (سیاه) و مقدار 255 نشان دهنده رنگ روشن (سفید) است.

به عنوان مثال تصویر زیر که سایز آن  $265 * 288$  است از یک ماتریس که دارای 288 سطر و 265 ستون است برای نمایش تصویر استفاده می‌کند.



هر پیکسل از این تصویر نیز مقداری بین 0 و 255 دارد. نقاط روشن مقادیری نزدیک به 255 و نقاط تیره مقادیر نزدیک به 0 دارد. همه توابع پردازش تصویر از این مقادیر استفاده کرده و اعمال لازم را بر روی تصویر انجام می‌دهند.

## کردن تصویر GrayScale

### تصویر Grayscale و RGB

یک تصویر RGB متشکل از سه تصویر است که هریک از آنها مقادیر Red ( قرمز ) ، Green ( سبز ) ، Blue ( آبی ) تصویر رنگی را نگه می دارند. تصویر حاصل نیز از ترکیب مقادیر درآیه های متناظر در سه ماتریس بوجود می آید.

$$F(x_1, y_1) = R(x_1, y_1) + G(x_1, y_1) + B(x_1, y_1)$$

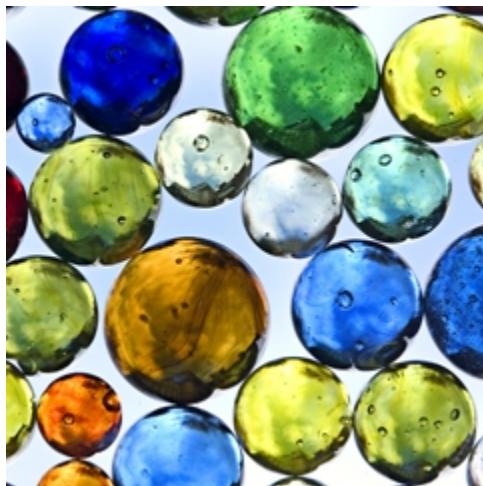
برای کردن تصویر از فرمول زیر استفاده می کنیم :

$$S(x_1, y_1) = (R(x_1, y_1) + G(x_1, y_1) + B(x_1, y_1)) / 3$$

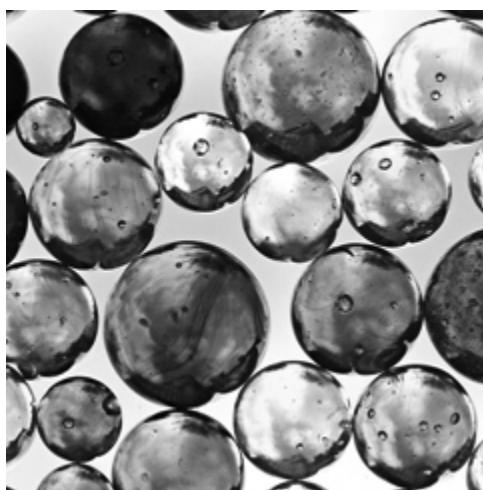
شبه کد زیر نحوه الگوریتم Grayscale کردن تصویر RGB با سایز  $N * M$  را نشان می دهد :

```
Procedure Grayscale( output , input As Bitmap )
Begin
    For I = 1 to M Do
        For J = 1 To N Do
            Temp = ( input.R( I,J ) + input.G( I,J ) + input.B( I,J ) ) / 3
            Output.R ( I,J ) = Temp
            Output.G ( I,J ) = Temp
            Output.B ( I,J ) = Temp
        End For
    End For
End
```

شکل زیر یک تصویر RGB را نشان می دهد :



شکل زیر نیز تصویر حاصل از اجرای الگوریتم **Grayscale** را نشان می دهد :



از آنجا که مقادیر ماتریس های **R** , **G** , **B** پس از **Grayscale** کردن تصویر باهم برابر هستند ، بنابراین می توان تنها از یک

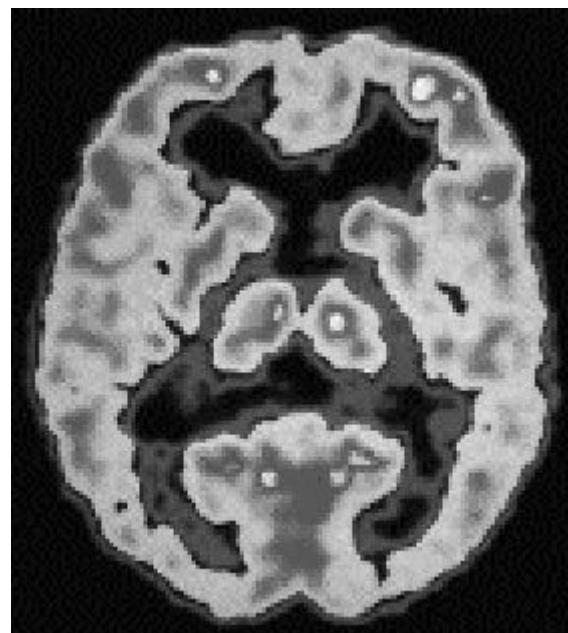
ماتریس برای نشان دادن مقادیر استفاده کرد. در الگوریتم های بعدی که برای پردازش تصویر ارائه خواهند شد از تصاویر

استفاده خواهیم کرد . بنابراین عملیات نیز بر روی تنها یک ماتریس انجام خواهد پذیرفت

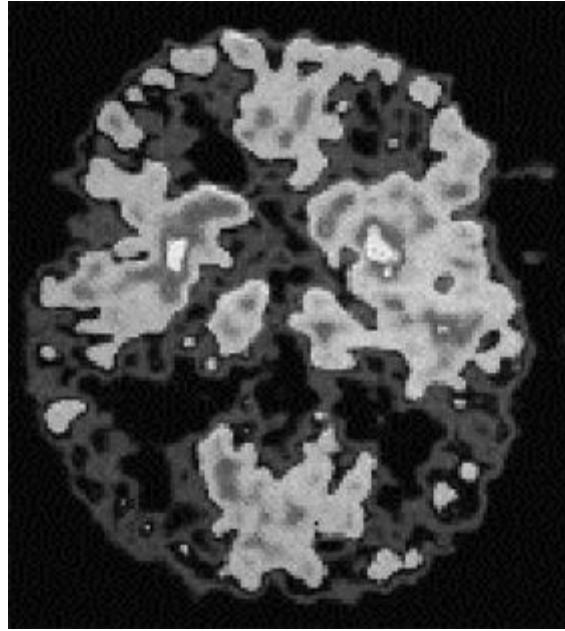
## تفریق دو تصویر ( Image Subtraction ) :

تفریق دو تصویر بدین مفهوم است که در دو تصویر ، شدت روشنایی پیکسل های متناظر دو تصویر را از هم کم کنیم . فرض کنید می خواهیم تغییراتی را که بر روی مغز افرادی که دچار بیماری آلزایمر هستند ، بررسی کنیم . برای این منظور می توانیم تصویری از یک مغز سالم را در مراحل مختلف با تصویر مغز فردی که دچار بیماری آلزایمر است ، مقایسه کنیم . با اعمال عملگر فوق بر روی دو تصویر مذکور نقاطی از مغز که در آن نقاط مغز دچار تغییر شده است مشخص می شوند .

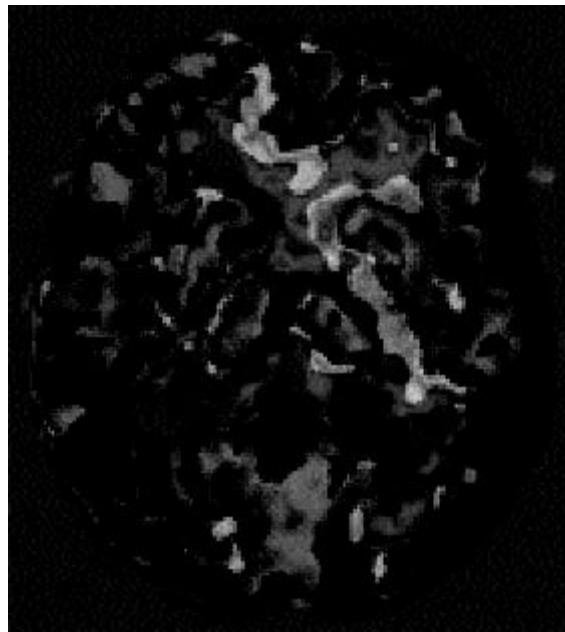
تصویر زیر عکس اسکن PET مغز نرمال را نشان می دهد :



تصویر زیر عکس اسکن PET مغز بیمار را نشان می دهد :



پس از اعمال عملگر تفریق نقاط تغییر یافته مغز مشخص می شوند :



شبه کد زیر نحوه پیاده سازی عملگر تفریق را نشان می دهد :

```
Procedure Subtract (image1, image2 : Byte[][])
```

```
Begin
```

```
    result :Byte[][]
```

```

sub:integer

For i = 1 To Height Do

    For j = 1 To Width Do

        sub = image1[i, j] - image2[i, j]

        result[i, j] = sub < 0 ? 0 : sub

    End For

End For


Return result;

End

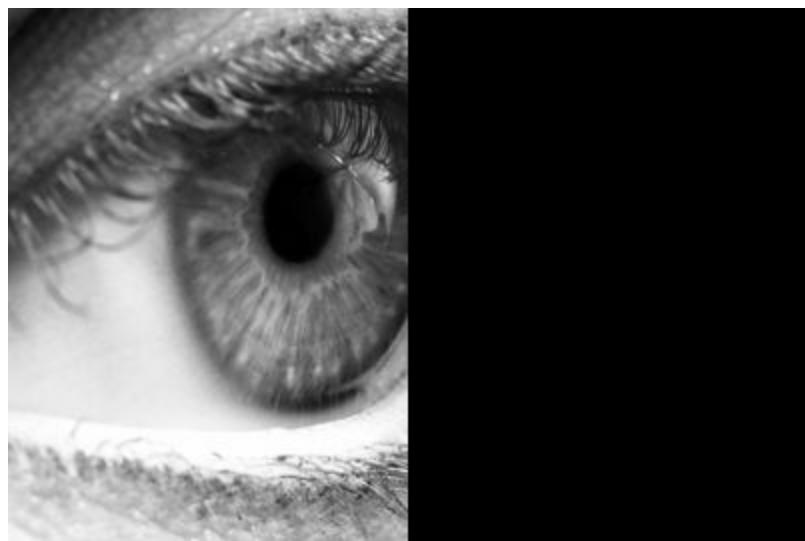
```

توجه : هنگام تفریق مقادیر پیکس ها ، مقادیر منفی را به مقدار صفر تبدیل می کنیم .

لازم به ذکر است که روش یاد شده برای تفریق دو تصویر ساده ترین روش تفریق است و دو روش دیگر نیز وجود دارد که تفریق دو تصویر در اغلب اوقات با استفاده از این روش ها انجام می پذیرد.

### جمع دو تصویر ( **Image Addition** ) :

جمع دو تصویر بدین مفهوم است که در دو تصویر ، شدت روشنایی پیکسل های متناظر دو تصویر را از باهم جمع کنیم . دو تصویر زیر را در نظر بگیرید :





پس از اعمال عملگر جمع بر روی این دو تصویر ، نتیجه ای که حاصل می شود ، تصویر زیر خواهد بود :



شبه کد زیر نحوه پیاده سازی عملگر جمع را نشان می دهد :

```
Procedure Add (image1, image2 : Byte[][])
```

```
Begin
```

```
    result :Byte[][]
```

```
    sub:integer
```

```
    For i = 1 To Height Do
```

```
        For j = 1 To Width Do
```

```
            sub = image1[i, j] + image2[i, j]
```

```
            result[i, j] = sub > 255 ? 255 : sub
```

```

End For
End For
Return result;
End

```

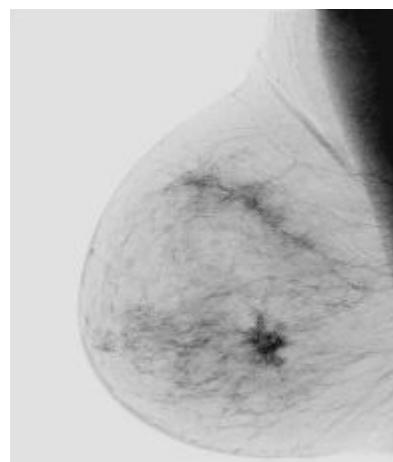
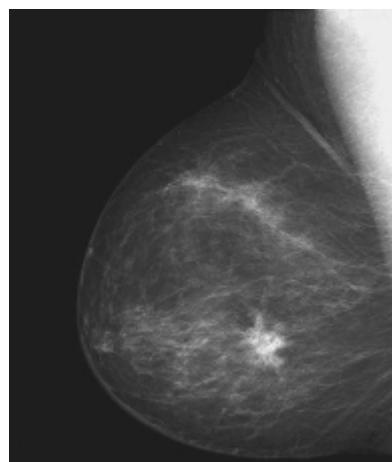
توجه : هنگام جمع مقادیر پیکس ها ، مقادیر بزرگتر از 255 را به مقدار 255 تبدیل می کنیم

## مکمل گردن تصویر

در یک تصویر سایه و سفید که فقط دو سطح شدت روشنایی 0 و 1 دارد ، منظور از مکمل کردن تصویر ، 1 کردن پیکسلهای با مقدار 0 و 0 کردن پیکسل های با مقدار 1 است. در تصویری که از L سطح روشنایی ( 0 .. L -1 ) برای نمایش شدت روشنایی پیکسل ها استفاده می کند ، مکمل یک پیکسل از فرمول زیر به دست می آید :

$$\text{Pixel}[ i , j ] = L - 1 - \text{Pixel}[ i , j ]$$

به شکل های زیر توجه کنید :



شکل سمت چپ یک تصویر ماموگرام است و شکل سمت راست نیز مکمل آن را نشان می دهد . همانطور که در این دو شکل به وضوح دیده می شود ، تشخیص اجزای تصویر مکمل شده آسانتر از تصویر اصلی است . در چنین مواردی که اجزای اصلی تصویر در یک تصویر با محدوده سیاه بزرگ قرار گرفته باشد، از مکمل تصویر استفاده می کنیم.

شبه کد زیر نحوه مکمل کردن تصویر را نشان می دهد :

```
Procedure Complement ( input : Image )
Begin
    output : Image
    For 1 To Height of image Do
        For 1 To Width of image Do
            output.pixels[ i , j ] = 255 - Input.pixels[ i , j ]
        End For
    End For
    return output
End
```

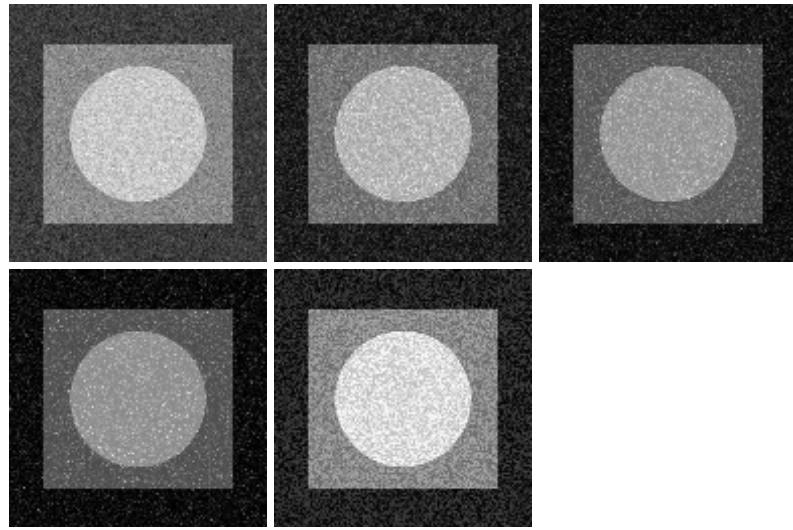
## میانگین گیری از تصویر

فرض کنید چند چند تصویر یکسان داریم که بر روی هر کدام از آنها نویزهای مختلفی وجود دارد و می خواهیم کیفیت این تصاویر را ارتقا دهیم. در چنین مواردی می توان از میانگین گیری از همه تصاویر استفاده کنیم. بدین صورت که مقادیر پیکسل های متناظر در همه تصاویر را باهم جمع کرده و سپس به تعداد تصاویر تقسیم کنیم. بدیهی است که هر قدر تعداد تصاویر برای میانگین گیری بیشتر باشد ، تصویر حاصل از میانگین گیری آنها نیز بیشتر به واقعیت نزدیک خواهد بود.

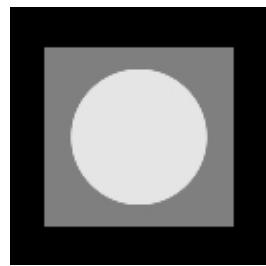
شبه کد زیر نحوه انجام این کار را برای دو تصویر نشان می دهد :

```
Procedure Average( bmp1 , bmp2 :Image)
Begin
    Result:Image
    For 1 To Height Do
        For 1 To Width Do
            Result.Pixels[ i , j ] = ( bmp1.Pixels[ i , j ] + bmp2.Pixels[ i , j ] ) /2
        End For
    End For
End
```

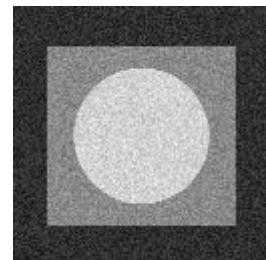
به عنوان مثال تصاویر نویز دار زیر را در نظر بگیرید :



تصویر بدون نویز نیز به صورت زیر است :



با میانگین گیری از تصاویر نویز دار ، تصویر زیر حاصل می شود :



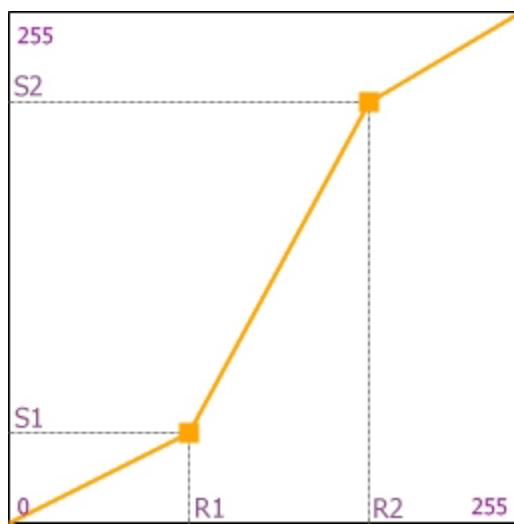
همانطور که مشاهده می کنید ، کیفیت تصویر حاصل از میانگین گیری ارتقا یافته و به تصویر بدون نویز نزدیک تر می شود

## کشش تمایز (Contrast Stretching)

در مواردی همچون تنظیم نبودن لنز و سیله تصویربرداری ، تصاویر با کنتراس پایین به وجود می آید. تصاویر کنتراس پایین

تصاویری هستند که در آن ها اختلاف کمترین و بیشترین سطح روشنایی کم است. برای ارتقای چنین تصاویری از تبدیل کشش

تمایز استفاده می کنیم. نمودار تبدیل کشش تمایز به در حالت کلی به صورت زیر است :



مقدار نقاط  $(r_1, s_1)$  و  $(r_2, s_2)$  (شکل نمودار را کنترل می کنند. در صورتی که  $r_1 = s_1$  و  $r_2 = s_2$  باشد ، تغییری در

تصویر حاصل نمی شود. برای این نمودار یک حالت خاص نیز وجود دارد که در این حالت  $s_2 = L-1$  ،  $r_1 = r_2$  و  $s_1 = 0$  است

حداکثر تعداد سطوح خاکستری است). به این حالت در اصطلاح آستانه گیری (Thresholding) گوییم که باعث می شود

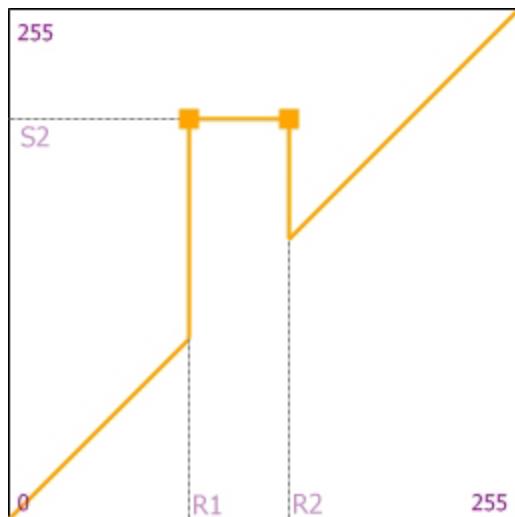
پیکسل های تصویری خروجی تنها شامل دو سطح باشند.

برای پیاده سازیتابع تبدیل کشش تمایز در داخلی ترین حلقه باید بر اساس نقاط  $(r_1, s_1)$  و  $(r_2, s_2)$  سه معادله خط

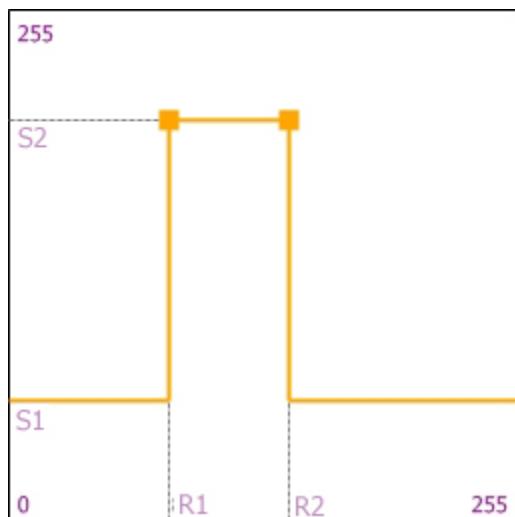
بنویسیم و سپس توسط آنها مقادیر  $s$  از تصویر ورودی را به مقادیر  $r$  از تصویر خروجی نگاشت کنیم

## بخش بندی سطح خاکستری (Gray-Level Slicing)

فرض کنید می خواهیم مساحت کل جنگل های استان را تعیین کنیم. با فرض اینکه عکس هوایی از استان را داشته باشیم ، می توانیم از بخش بندی سطح خاکستری استفاده کنیم. بدین صورت که با نمایان کردن سطوح خاکستری نشان دهنده جنگل ها و پایین آوردن شدت سطوح خاکستری دیگر نقاط تصویر ، مساحت جنگل ها را محاسبه کنیم. نمودار زیر نحوه نگاشت مقادیر پیکسل های تصویر ورودی را نشان می دهد.



همچنین از نمودار زیر نیز برای نگاشت مقادیر پیکسل های تصویر ورودی می توان استفاده کرد.



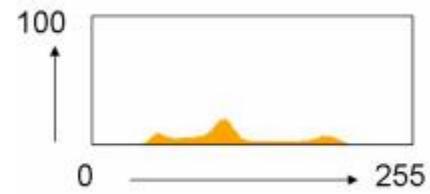
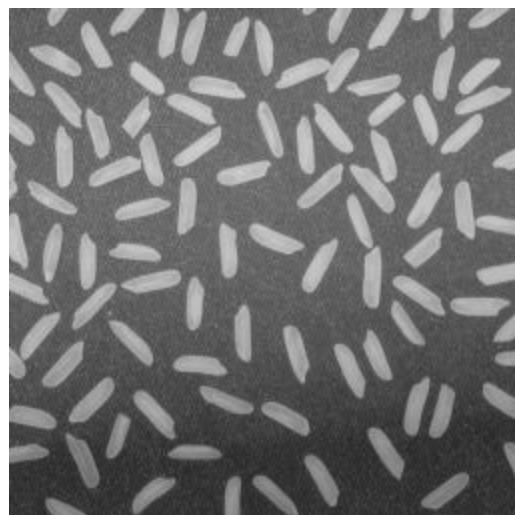
شکل نمودار فوق نشان می دهد که تصویر خروجی تولید شده توسط این نمودار یک تصویر باینری است.

## هیستوگرام تصویر (Histogram)

نمودار هیستوگرام تصویر نموداری است که در آن تعداد پیکسل های هر سطح روشنایی در تصویر ورودی مشخص می شود. فرض کنید تصویر ورودی یک تصویر Grayscale با 256 سطح روشنایی باشد ، بنابراین هریک از پیکسل های تصویر در شدت روشنایی در بازه  $[0..255]$  می توانند داشته باشند. برای به دست آوردن هیستوگرام تصویر ، با پیمایش پیکسل های تصویر ، تعداد پیکسل های هر سطح روشنایی را محاسبه می کنیم .

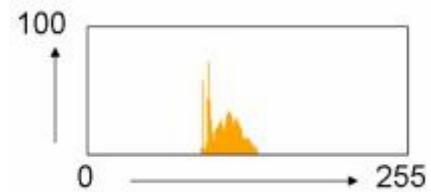
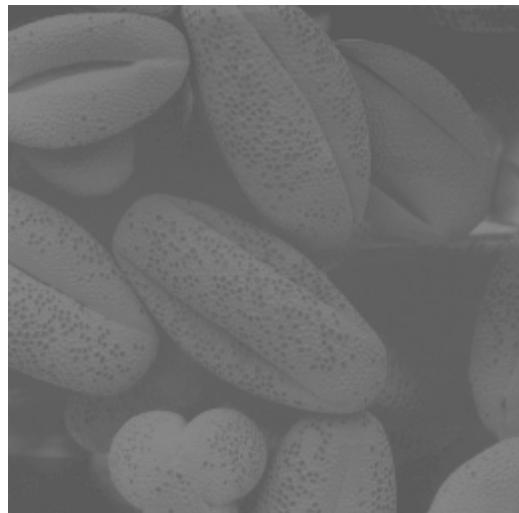
هیستوگرام نرمال نیز از تقسیم کردن مقادیر هیستوگرام به تعداد کل پیکسل های تصویر به دست می آید. نرمال سازی هیستوگرام

موجب می شود که مقادیر هیستوگرام در بازه  $[0,1]$  قرار گیرند. شکل زیر تصویری را به همراه هیستوگرام نرمال آن نشان می دهد :

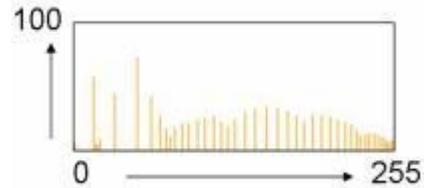
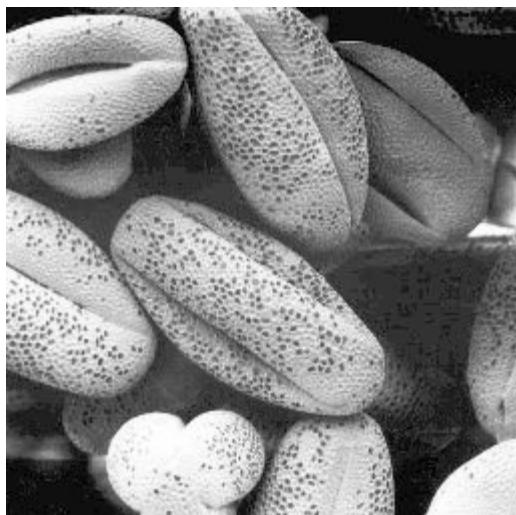


## تعدیل هیستوگرام ( Histogram Equalization )

به تصویر زیر توجه کنید :



همانطور که از شکل و هیستوگرام تصویر مشخص است ، کنتراست تصویر فوق پایین می باشد. بدین مفهوم که اختلاف کمترین و بیشترین شدت روشنایی کم است. با تغییل سازی هیستوگرام تصویر با کنتراست پایین ، تصویری با کنتراست بالا به دست می آید :



شکل فوق تصویر قبلی پس از اجسام تعدیل سازی هیستوگرام را نشان می دهد.

الگوریتم زیر روش تعدیل سازی هیستوگرام را نشان می دهد :

1) هیستوگرام تصویر را محاسبه می کنیم. فرض کنید مقادیر هیستوگرام در آرایه  $hist$  قرار گیرد.

2) با استفاده از فرمول زیر فراوانی هیستوگرام را محاسبه می کنیم :

$$histCum[i] = histCum[i - 1] + hist[i]$$

3) از فرمول زیر استفاده کرده و هیستوگرام تعدیل شده را محاسبه می کنیم :

$$eqHist[i] = \text{Truncate}([(L * histCum[i]) - N]/N)$$

که در این فرمول  $L$  تعداد سطوح خاکستری و  $N$  تعداد کل پیکسل ها را نشان می دهد

4) در مرحله نهایی مقادیر جدید پیکسل ها را به صورت زیر مقدار دهی می کنیم :

$$Result[i, j] = eqHist[input[i, j]]$$

که  $Result$  تصویر خروجی و  $input$  تصویر ورودی را نشان می دهد

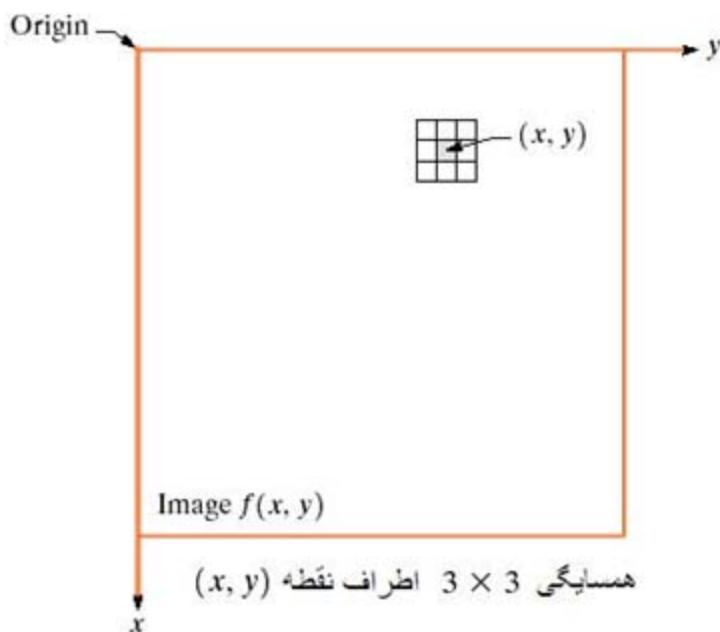
## ارتقای تصویر (Image Enhancement)

ارتقای تصویر در دو حوزه مکانی و فرکانسی انجام می‌پذیرد که تصویر در حوزه مکانی با استفاده از مقادیر پیکسل های همسایه و در حوزه فرکانسی توسط بسط فوریه تصویر دیجیتالی، تصویر ارتقا می‌یابد. در این بخش ارتقای تصویر در حوزه مکانی را مورد بررسی قرار می‌دهیم.

ارتقای تصویر در حوزه مکانی با استفاده از فرمول زیر انجام می‌گیرد:

$$S = T(r)$$

که در این فرمول  $S$  تصویر ارتقا یافته،  $r$  تصویر ورودی و  $T(\cdot)$  تابعی است که بر روی تصویر اعمال می‌شود. در حوزه مکانی منظور از  $T(\cdot)$  یک ماسک  $n * n$  است که بر روی همه پیکسل‌های تصویر اعمال می‌شود. شکل زیر تصویر و ماسک اعمالی بر روی تصویر در نقطه  $(x, y)$  را نشان می‌دهد.



در واقع ماسک را می توان پنجره ای در نظر گرفت که بر روی همه پیکسل های تصویر حرکت کرده و در هر نقطه با توجه به مقادیر

MASK ، مقدار شدت روشنایی پیکسل مرکزی ( نقطه  $y, x$  در شکل ) محاسبه می شود. البته مقادیری که در ماسک قرار می

گیرند ، ضرایبی می باشند که نشان می دهند هر یک از پیکسل های همسایه تا چه حد در تعیین مقدار پیکسل مرکزی تاثیر گذارند.

برای محاسبه مقدار پیکسل تصویر خروجی در نقطه  $(y, x)$  با استفاده از یک ماسک  $3 \times 3$  از فرمول زیر استفاده می کنیم  $W$

ماتریس  $3 \times 3$  ماسک و  $f$  ماتریس  $N \times M$  تصویر است :

$$G(x, y) = w(x-1, y-1) * f(x-1, y-1) + w(x-1, y) * f(x-1, y) + w(x-1, y+1) * f(x-1, y+1) \\ + w(x, y-1) * f(x, y-1) + w(x, y) * f(x, y) + w(x, y+1) * f(x, y+1) \\ + w(x+1, y-1) * f(x+1, y-1) + w(x+1, y) * f(x+1, y) + w(x+1, y+1) * f(x+1, y+1)$$

در این رابطه  $w(i, j)$  مقدار ماسک در نقطه  $(j, i)$  و  $f(i, j)$  مقدار پیکسل تصویر در نقطه  $(j, i)$  را نشان می دهد.

پیکسل تصویر خروجی در نقطه  $(j, i)$  را حركت دادن ماسک بر روی تصویر و محاسبه مقدار هر یک از پیکسل

ها با استفاده از فرمول یاد شده ، تصویر خروجی حاصل می شود. بخش بعدی به بررسی برخی از فیلترهای حوزه مکانی پرداخته

است.

## فیلترهای حوزه مکانی

برای انجام تبدیلات در هر دو حوزه مکانی از فیلترها استفاده می کنیم. فیلترها در به دو دسته تقسیم می شوند:

• فیلترهای پایین گذر ( Low Pass Filter ) : که با عبور از پیکسل های با فرکانس کم ، بر روی

پیکسل های با فرکانس بالا تغییرات ایجاد می کنند.

فیلترهای بالاگذر ( High Pass Filter ) : که با عبور از پیکسل های با فرکانس بالا ، بر روی پیکسل

های با فرکانس پایین تغییرات ایجاد می کنند.

توضیح اینکه پیکسل فرکانس پایین پیکسلی است که اختلاف شدت روشنایی آن پیکسل با پیکسل های همسایه

اش کم باشد ( نقاط آرام تصویر ) . در مقابل پیکسل فرکانس بالا پیکسلی است که اختلاف شدت روشنایی آن

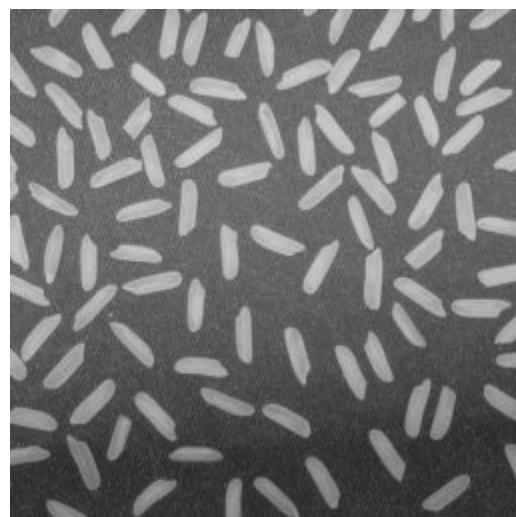
پیکسل با پیکسل های همسایه اش زیاد باشد ( لبه ها و نویزها ).

با توجه به مطالب فوق می توان نتیجه گرفت که نتیجه اعمال فیلترهای پایین گذر ، تصویری آرام خواهد بود ( لبه

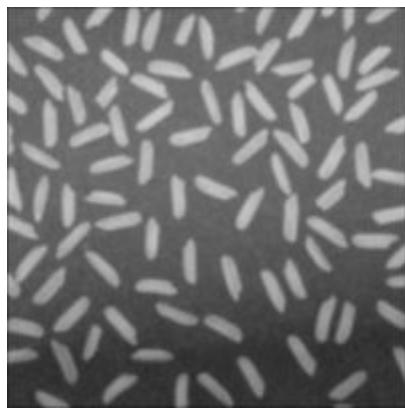
ها و نویزها تا حدودی آرام می شوند ) و با اعمال فیلتر بالاگذر نیز تصویری شامل لبه ها و نویزهای تصویر به دست

می آید ( نقاط آرام به صورت پیکسل های سیاه رنگ نمایان می شوند ).

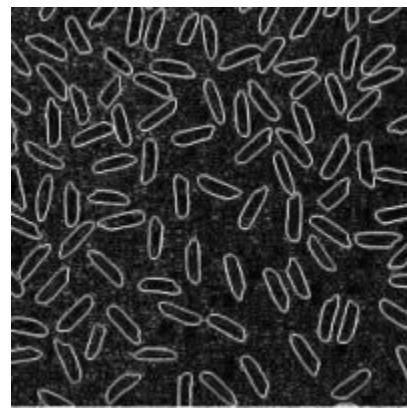
شکل زیر تصویری را پس از اعمال فیلتر پایین گذر و بالا گذر نشان می دهد :



تصویر اصلی



اعمال فیلتر پایین گذر



اعمال فیلتر بالاگذر

فیلترهای پایین گذر:

فیلترهای میانگین:

ساده ترین نوع فیلترهای پایین گذر فیلترهای میانگین می باشند. یک فیلتر میانگین  $n * m$  شامل ضرایبی مثبت می باشند که همه این ضرایب بر عکس مجموع کل ضرایب فیلتر ضرب می شوند. به عنوان مثال فیلتر  $3 * 3$  زیر را در نظر

بگیرید:

$$(1/9) * \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

فیلتر فوق یکی از فیلترهای میانگین پرکاربرد در پردازش تصویر است که همه ضرایب آن 1 بوده و مجموع ضرایب نیز 9 می باشد.

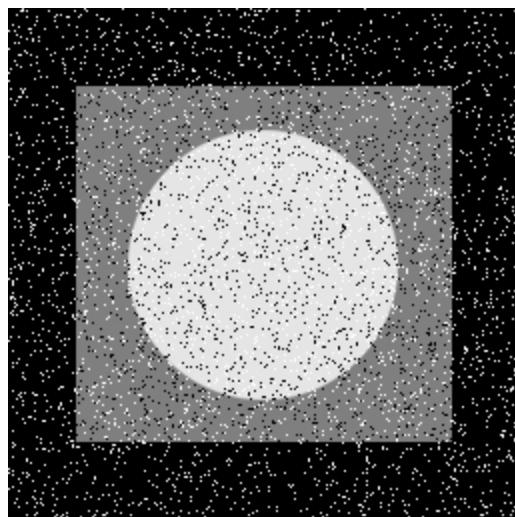
برای بلور کردن بیشتر تصویر از ماسک های بزرگتری با سایز  $n * m$  باید استفاده کرد که مقادیر همه عناصر ماسک 1 بوده و کل ماسک به  $1/mn$  ضرب شود. بدیهی است هرچه اندازه ماسک بزرگتر باشد ، زمان بیشتری برای پردازش تصویر نیاز خواهد بود.

در زیر فیلتر آرام کننده دیگری که به فیلتر پایین گذر گاووس ( Gaussian ) مشهور است ، نمایش داده شده است :

$$(1 / 16) * \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 4 & 1 \\ \hline \end{array}$$

### ( Median ) فیلتر میانه

فیلتر پایین گذر دیگری که نسبت به فیلترهای دیگر به زمان پردازش بیشتری نیاز دارد ، فیلتر میانه است. فیلتر پایین گذر میانه از یک همسایگی  $n * m$  استفاده می کند و روش کار آن نیز به این صورت است که کل همسایگی ها را به صورت صعودی مرتب کرده و عنصر وسط اعداد مرتب شده را انتخاب و جایگزین پیکسل مرکزی می کند. لازم به ذکر است که فیلتر پایین گذر میانه برای حذف نویز فلفل نمکی ( Salt & Pepper ) می تواند مورد استفاده قرار گیرد. شکل زیر تصویر نویزداری ( نویز فلفل نمکی ) را نشان می دهد که با استفاده از فیلتر میانه ارتقا یافته است:



تصویر اصلی



تصویر فیلتر شده

### فیلترهای بالاگذر

فیلترهای بالاگذر معمولاً از دو ماسک برای تشخیص لبه‌ها استفاده می‌کنند. فرض کنید مقدار محاسبه شده از ماسک اول  $p$  و

مقدار محاسبه شده از ماسک دوم  $q$  باشد. مقدار پیکسل مرکزی پس از اعمال هر دو ماسک از فرمول زیر به دست می‌آید:

$$G = \text{ABS}(p) + \text{ABS}(q)$$

یا

$$G = \text{SQRT}((p*p) + (q*q))$$

### : ( Prewitt ) فیلتر پرویت

ماسک‌های فیلتر پرویت به صورت زیر هستند:

1	1	1
0	0	0
-1	-1	-1

1	0	-1
1	0	-1
1	0	-1

## فیلتر سوبل (Sobel) :

ماسک های فیلتر سوبل به صورت زیر هستند:

1	2	1
0	0	0
-1	-2	-1

1	0	-1
2	0	-2
1	0	-1

## فیلتر Frei-Chen :

ماسک های فیلتر Frei-Chen به صورت زیر هستند:

1	SQRT(2)	1
0	0	0
-1	-SQRT(2)	-1

1	0	-1
SQRT(2)	0	-SQRT(2)
1	0	-1

## فیلتر لاپلاس (Laplacian) :

بر خلاف فیلتر های سوبل ، پرویت و Frei-Chen ، فیلتر لاپلاسین یک فیلتر مشتق مرتبه دوم می باشد. از این رو تنها نیاز به یک

ماسک برای مشخص کردن لبه ها دارد :

## ارتقای تصویر در حوزه فرکانسی ( Frequency Domain )

ارتقای تصویر در حوزه فرکانسی با استفاده تبدیل فوریه تصویر انجام می گیرد. روش کلی ارتقای تصویر در حوزه فرکانسی بدین

صورت است :

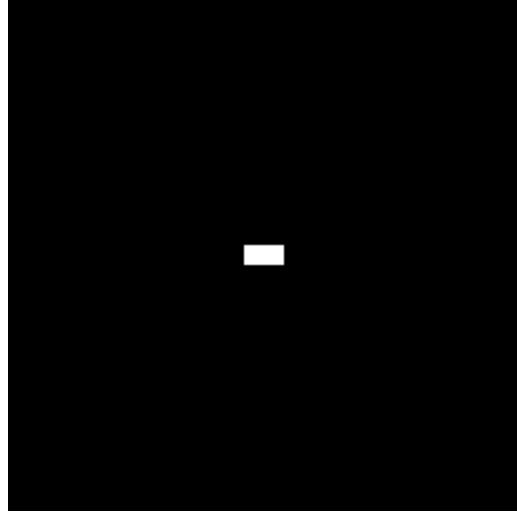
1. ابتدا تبدیل فوریه تصویر را محاسبه می کنیم ( شیفت دادن مرکز فرکانس در همین قسمت انجام می پذیرد )
2. فیلتر مورد نظر را بر روی تبدیل فوریه اعمال می کنیم
3. تبدیل فوریه معکوس تصویر را محاسبه کرده و تصویر فیلتر شده را به دست می آوریم ( شیفت دادن دوباره مرکز فرکانس در همین قسمت انجام می پذیرد )

تبدیل فوریه تصویر دیجیتالی به طور کامل در کتاب آقای Digital Image Processing Rafel C. Gonzalez و

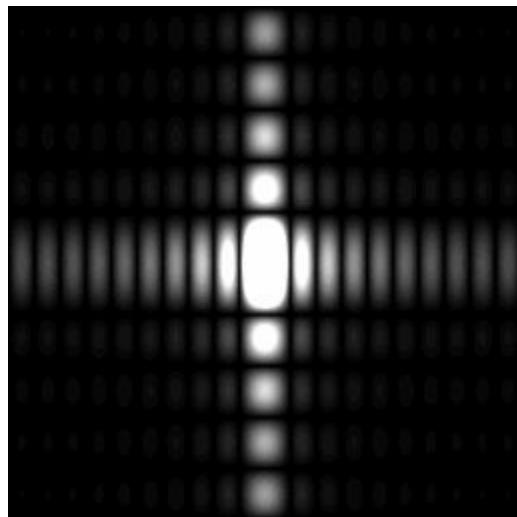
Richard E. Woods شرح داده شده است.

در حالت کلی در تصویر دو نوع پیکسل وجود دارد ، پیکسل های آرام و پیکسل های تیز . پیکسل های آرام پیکسل هایی هستند که اختلاف شدت روشنایی آن ها با شدت روشنایی پیکسل های همسایه شان کم است. در مقابل پیکسل های تیز ، پیکسل هایی هستند که اختلاف شدت روشنایی آن ها با شدت روشنایی پیکسل های همسایه شان زیاد است ( مانند لبه ها و نویز ها ) .

به عنوان مثال در تصویر زیر پیکسل های قرار گرفته در محیط مستطیل ، پیکسل های تیز و بقیه پیکسل های آرام هستند.



تبديل فوريه شكل فوق به صورت زير است :



در شكل حاصل از تبديل فوريه تصوير دو نكته وجود دارد :

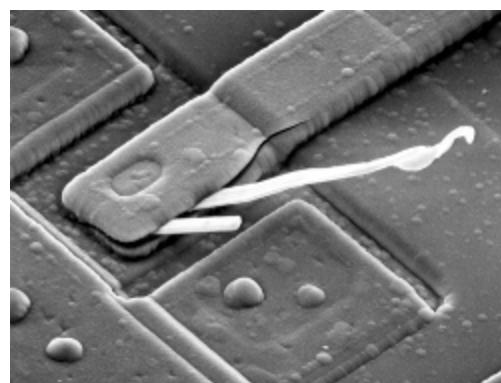
1. هرچه به مرکز تبديل فوريه نزديك می شويم ، پикسل ها آرام تر می شوند

2. هرچه به کناره های تبديل فوريه نزديك شويم ، پикسل های تندرتر می شوند.

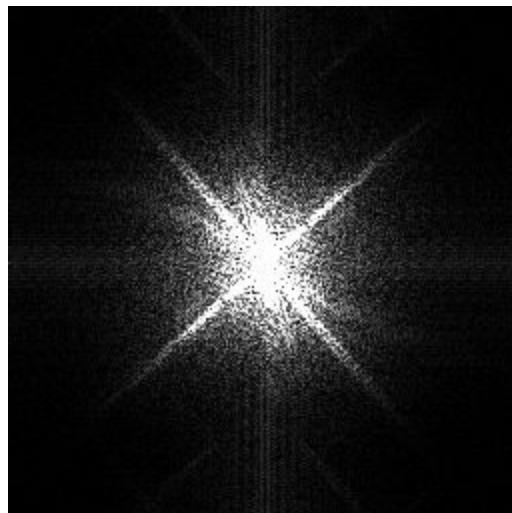
به نظر شما در تصويری با نويز بالا شكل بسط فوريه چگونه خواهد بود ؟ در انتهای اين بخش جواب اين سوال را بررسی خواهيم کرد.

از نکات فوق می توان چنین نتیجه گرفت که دستکاری پیکسل های مرکزی تبدیل فوریه همانند دستکاری پیکسل های آرام تصویر ، و تغییر دادن پیکسل های کناری تبدیل فوریه مشابه دستکاری پیکسل های تیز تصویر می باشد. دو نکته مطرح شده ، هنگام طراحی فیلتر ها بسیار استفاده می شوند.

به عنوان مثال به شکل زیر و تبدیل فوریه آن توجه کنید :

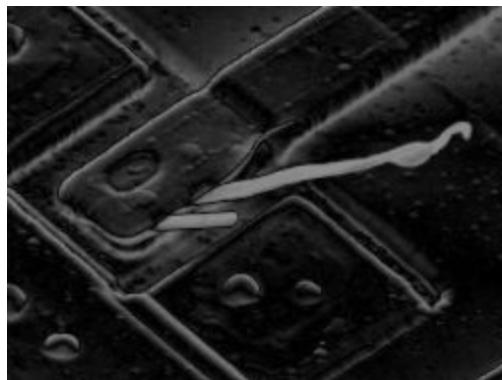


تصویر اصلی



بسط فوریه تصویر

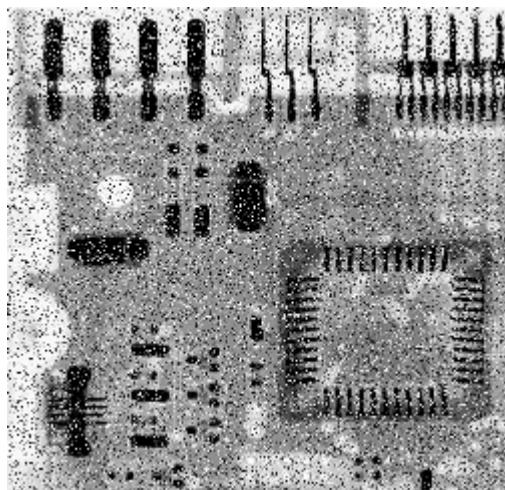
حال مقدار پیکسل مرکزی تبدیل فوریه را به مقدار  $0$  ضرب کرده و تبدیل فوریه معکوس آن را محاسبه می کنیم :



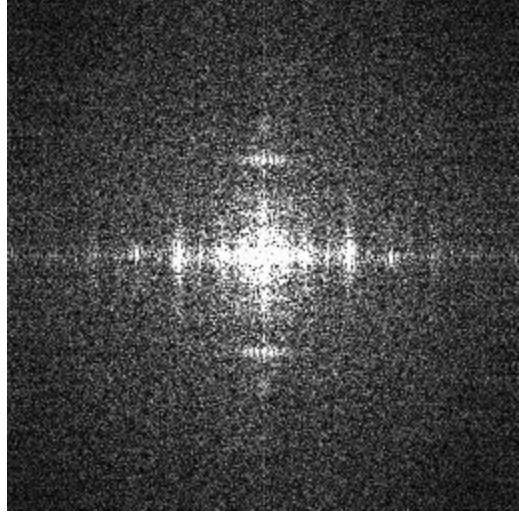
همانطور که مشاهده می کنید ، بیشتر پیکسل های آرام از تصویر حذف شده اند. در مقاله بعد با استفاده از همین خصوصیات فیلتر های بالا گذر و پایین گذر را بررسی خواهیم کرد.

حال به سوالی که در بالا مطرح شده است ، می پردازیم . سوال این است که در یک تصویر با نویز بالا ف شکل تبدیل فوریه چگونه خواهد بود ؟ این سوال را به همراه مثالی بررسی می کنیم.

تصویر نویز دار زیر را در نظر بگیرید :

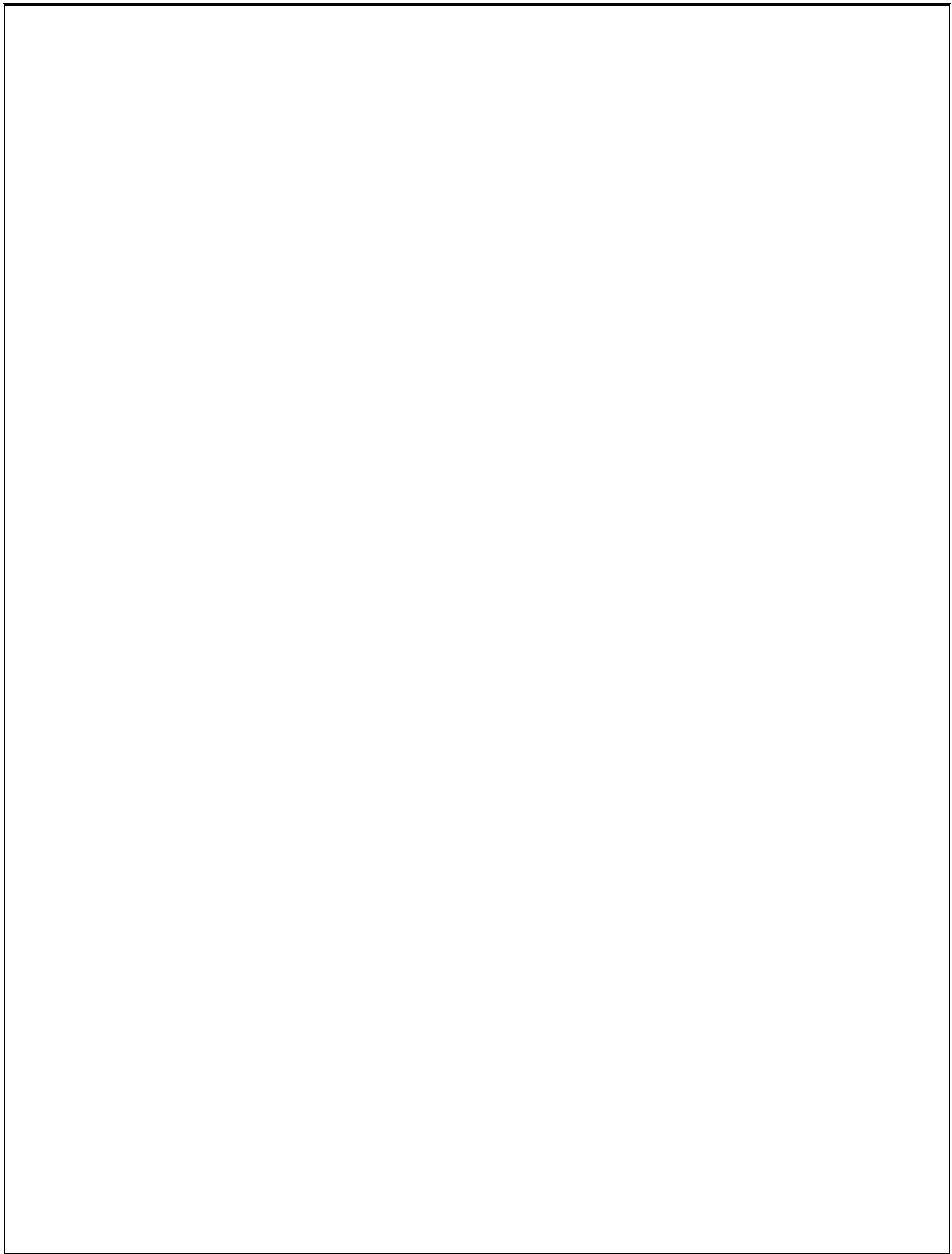


تبدیل فوریه این تصویر به صورت زیر است :



همانطور که مشاهده می کنید ، در تبدیل فوریه تصویر نویز دار به علت اینکه هر نویز یک پیکسل تیز می باشد ، بنابراین کناره های تبدیل فوریه به صورت دانه های سفید رنگ نمایش می یابند.

همانطور که مشاهده می کنید ، در تبدیل فوریه تصویر نویز دار به علت اینکه هر نویز یک پیکسل تیز می باشد ، بنابراین کناره های تبدیل فوریه به صورت دانه های سفید رنگ نمایش می یابند



فصل سوم

# پردازش تصویر فازی

## چرا پردازش تصویر فازی ؟

بسیاری از آدمها و نه فقط مخالفان منطق فازی، می‌پرسند چرا باید از تکنیک‌های فازی در پردازش تصویر استفاده کرد.

البته دلایل بسیاری برای این قضیه وجود دارد که مهمترین آنها به شرح زیر است:

- 1 - تکنیک‌های فازی ابزارهای قدرتمندی برای نمایش و پردازش دانش هستند.
- 2 - تکنیک‌های فازی قابلیت مدیریت ابهام و گنجی را به خوبی دارا هستند.

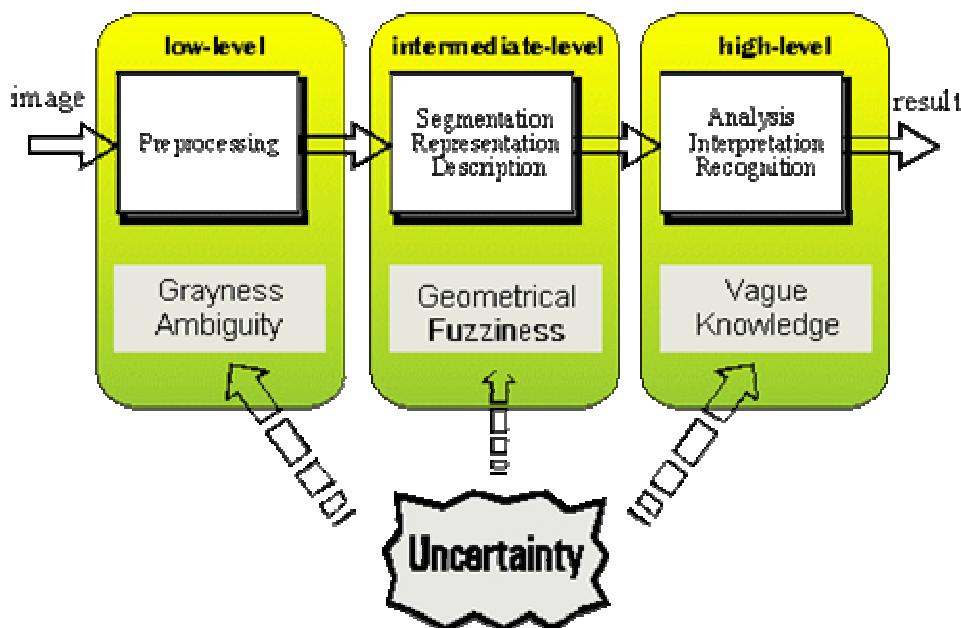
در بسیاری از کاربردها پردازش تصویر، نیازمند به استفاده از دانش کارشناسان خبره می‌باشیم تا قادر باشیم بر سختی‌های کار (مانند تشخیص شیء، تحلیل منظره) فائق آئیم. تئوری مجموعه‌های فازی و منطق فازی امکان نمایش و پردازش اطلاعات بشری را در ساختار قوانین IF- THEN در اختیار قرار می‌دهند. از سوی دیگر، بسیاری از سختی‌ها در پردازش تصویر ناشی از عدم قطعیت داده‌ها، وظایف و نتایج می‌باشد. این عدم قطعیت به معنای دارا بودن حالت تصادفی نیست بلکه به معنی ابهام و گنجی ذاتی مسئله می‌باشد. در کنار حالت‌های تصادفی که می‌توانند به کمک تئوری احتمال مدیریت گردند ما قادر به تشخیص و تمیز ما بین 3 نوع نقص در پردازش تصویر خواهیم بود:

- ابهام GRAYSCALE

- فازیت هندسی

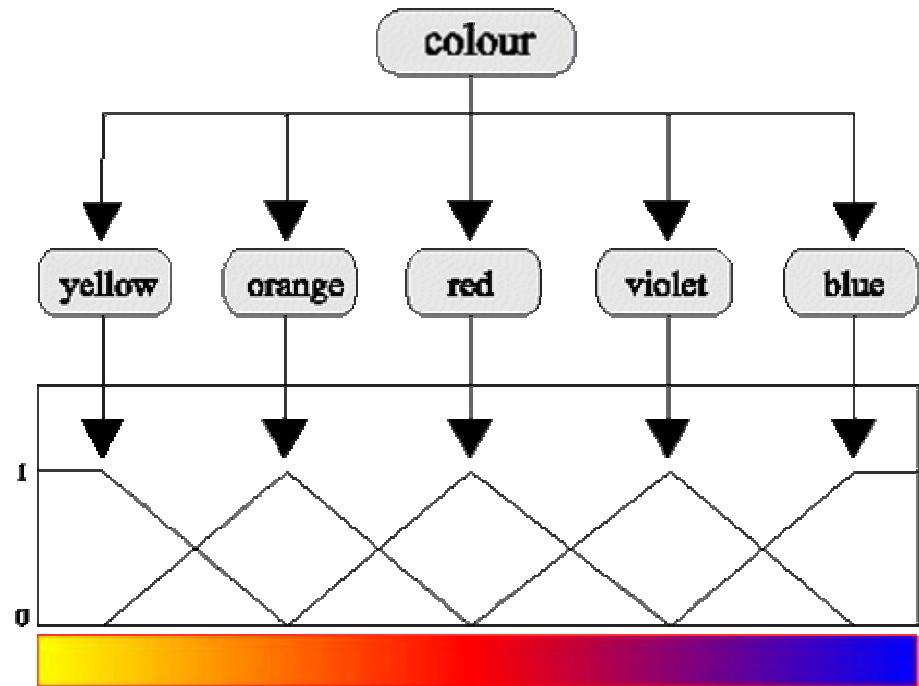
- دانش مبهم(پیچیده/ بد تعریف)

این مسائل طبیعتی فازی دارند. مسائل زیر و نمونه‌ها مشابه مثال‌های مناسبی‌اند که در آن‌ها استفاده از تکنیک‌های فازی کارایی بیشتری در جهت مدیریت نقص‌های پردازش فازی معمولی دارند.



< نقص و عدم قطعیت در پردازش تصویر >

برای مثال ما می‌توانیم به رنگ‌های مختلف به عنوان مجموعه فازی نگاه کنیم که دارای اعضای زرد، نارنجی، قرمز، بنفش و آبی می‌باشد. مرزهای میان رنگ‌ها را در فازی به گونه‌ای بهتر می‌توان نمایش داد و بدین ترتیب محاسبات جزئی‌تر ممکن خواهد شد.



< نمایش رنگ‌ها به عنوان مجموعه‌ای فازی. >

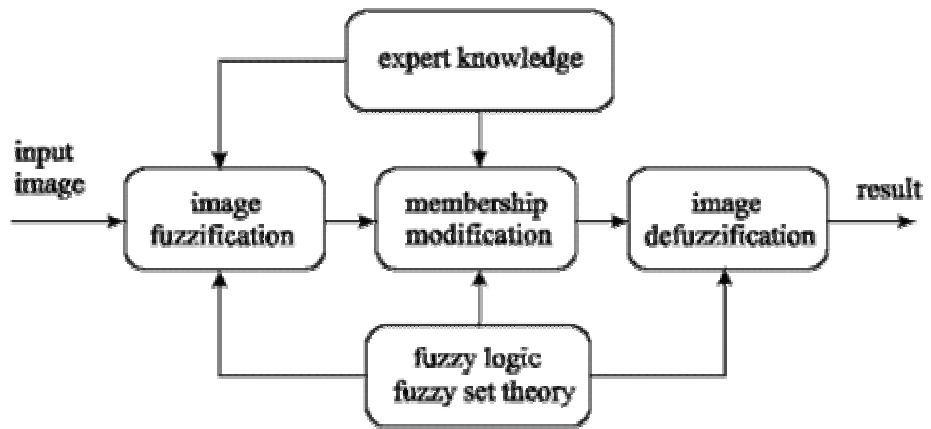
## پردازش تصویر فازی یعنی چه؟

پردازش تصویر فازی یک نظریه تنها نیست بلکه مجموعه‌ای از رویکردهای فازی مختلف در پردازش تصویر است. با این حال تعریف زیر را می‌توان ابزار مناسبی برای تعیین مرزهای کاری در نظر گرفت:

پردازش تصویری فازی مجموعه‌ای از رویکردها و دیدگاه‌هایی است که با نگرش فازی نسبت به بخش‌ها و ویژگی‌های تصویر آن را فهمیده، نمایش داده و پردازش می‌کند که این نمایش و پردازش بستگی به تکییک‌های فازی انتخاب شده و ماهیت مسئله دارد.

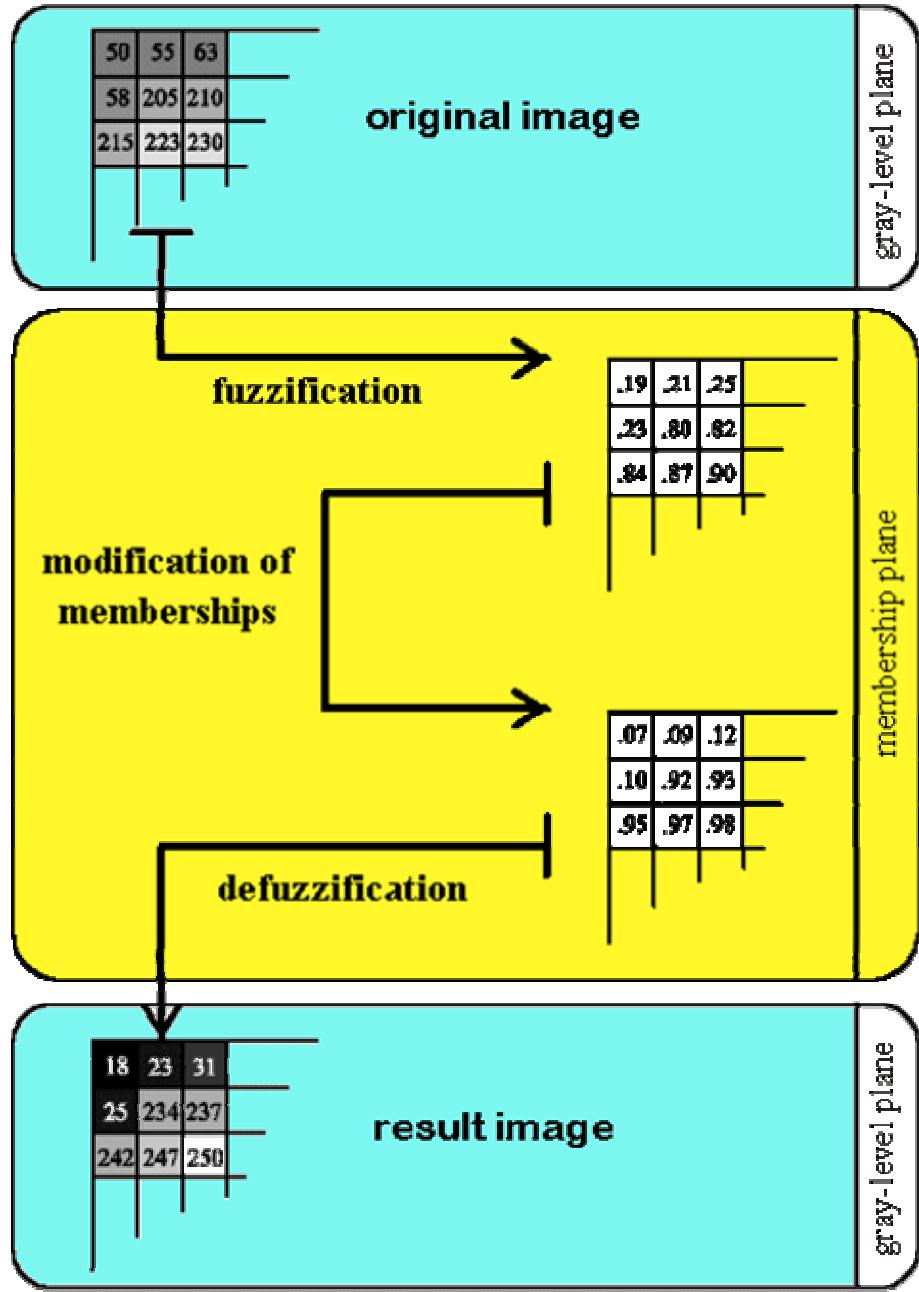
پردازش تصویر فازی 3 مرحله اصلی دارد:

فازی کردن تصویر، اصلاح ارزش‌های عضویت، در صورت نیاز از حالت فازی خارج کردن.



مراحل فازی کردن تصویر و از حالت فازی خارج کردن از آنجا ناشی می‌شود که ما روی سختافزار فازی پردازش نمی‌کنیم. بنابراین کدگذاری داده‌های تصویری(فازی کردن) و رمزگشایی نتیجه(از فازی خارج کردن) گام‌هایی هستند که پردازش روی تصویرها را با تکنیک‌های فازی ممکن می‌سازند.

قدرت اصلی پردازش تصویر فازی در مرحله میانی(اصلاح ارزش‌های عضویت) است. پس از آنکه داده‌های تصویر از لایه‌ای gray-level به لایه‌ی عضویت تغییر شکل داد(فازی کردن)، تکنیک‌های فازی مناسب ارزش‌های عضویت را اصلاح می‌کنند. این می‌تواند یک دسته‌بندی فازی، یک رویکرد قانون‌گرایی فازی، یک دیدگاه یکپارچه‌سازی فازی و ... باشد.



## تئوری پردازش تصویر فازی

پردازش تصویر فازی مجموعه‌ای از حوزه‌های مختلف تئوری مجموعه فازی، منطق فازی و تئوری اندازه‌گیری فازی است. موضوعات زیر بیانگر مهمترین مولفه‌های نظری و پردازش تصویر فازی هستند:

- \* هندسه فازی(متريک، توپولوژي)
  - \* اندازه‌گیری‌های ميزان فازی بودن و اطلاعات تصویر
  - \* سистем‌های استنتاج فازی(فازی کردن تصویر، استنتاج، از حالت فازی خارج کردن)
  - \* دسته‌بندی فازی(C-means فازی، C-means احتمالی)
  - \* ریخت‌شناسی ریاضیاتی فازی(سایش فازی، گسترش فازی)
  - \* تئوری اندازه‌گیری فازی
  - \* گرامرهای فازی
  - \* مباحث تركیبی(شبکه‌های عصبی فازی، الگوریتم‌های زنتیک فازی، تحلیل موجی فازی)
  - \* بسط روش‌های کلاسیک(تبديل Hough فازی، فیلتر میانه فازی)
- ما برای اینکه قادر باشیم از این تکنیک‌های گفته شده به صورت سیستماتیک استفاده کنیم لازم است که به درک تصویری مناسبی دست پیدا کنیم. فازی کردن تصویر نقش محوری ای در همه سیستم‌های پردازش تصویر، برای به کار بستن این مولفه‌ها به عهده دارد.

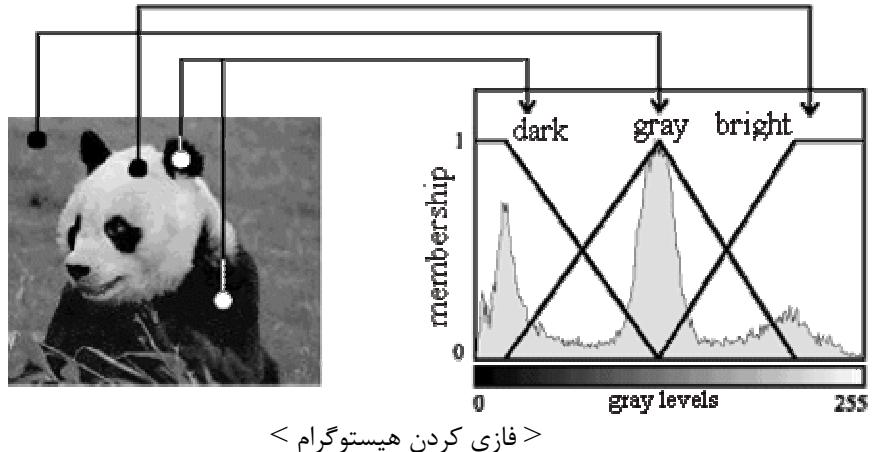
سه نوع مختلف از فازی کردن تصویر وجود دارد:

1. فازی کردن gray-level (يا فازی کردن هیستوگرام به صورت مختصر)

مثال: روشنی در ارتقای تصویر(شکل اول)

2. فازی کردن محلی(مثال: E. D. لبه‌یابی)

3. فازی کردن ویژگی(تجزیه تبدیل منظره A, S, شناخت شیء R (O.



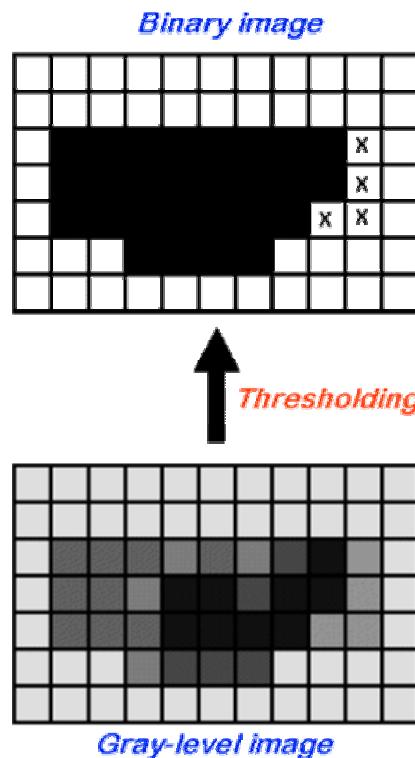
< فازی کردن هیستوگرام >

هندسه دیجیتالی نقش مهمی در بسیاری از کاربردهای پردازش تصویر به عهده دارد. عموماً تصاویر Gvay-level برخی پارامترهای هندسی نظری مساحت، محیط، قطر، فشردگی آستانه‌گیری می‌شوند. اما هنگامی که تصویر یا بخش‌های آن بدتریف یا غیر قطعی اند در نظر گرفتن آن به عنوان مجموعه‌های فازی منطقی و کاراست. آقای Azriel Rosarfeld برای نخستین بار مفهوم هندسه دیجیتالی فازی را معرفی نمود:

راه منطقی و استاندارد برای آنالیز و شناسایی یک تصویر بخش‌بندی آن به نواحی مختلف، محاسبه ویژگی‌های هر ناحیه و یافتن روابط بین نواحی مختلف آن است.

اما این نواحی همیشه خوش تعریف و قطعی نیستند. به همین علت گاهی بهتر است آن‌ها را به عنوان زیر مجموعه‌های فازی یک تصویر در نظر بگیریم. تعیین ویژگی‌های هندسی برای یک مجموعه فازی همواره آشکار و آسان نیست. اما تعاریف صریح بوده و ویژگی‌های پایه‌ای برای انواع مختلف این ویژگی‌ها و روابط (شامل همبندی، محاط بودن، محدب بودن، مساحت محیط، فشردگی، اندازه و قطر) به کار گرفته شده‌اند. (هندسه فازی برای زیر مجموعه‌های تصویر؛ نوشته: (A. Rosenfeld

برای مثال برای محاسبه کشیدگی یک شیء آستانه‌گیری همواره بهینه نیست (ممکن است منجر به از دست رفتن برخی اطلاعات گردد) بنابراین، بهتر است اعمال را روی تصویر gray-level اصلی اعمال کرده و به چنین تقسیم بندی تن در ندهیم.



اگر تابع عضویت مناسبی برای شی مورد نظر بیاییم مساحت  $a$  و محیط  $p$  در آن شی به راحتی قابل محاسبه خواهد بود:

$$a(\mu) = \sum \mu -$$

مساحت مجموعه‌های فازی:

محیط یک مجموعه فازی:

$$p(\mu) = \sum_{m=1}^M \sum_{n=1}^{N-1} |\mu_{mn} - \mu_{m,n+1}| + \sum_{n=1}^N \sum_{m=1}^{M-1} |\mu_{mn} - \mu_{m+1,n}|.$$

بر پایه مساحت و محیط می‌توان فشردگی یک مجموعه فازی را مانند زیر تعیین کرد.

فرمول:

$$\text{Compactness}(\mu) = \frac{a(\mu)}{p^2(\mu)}.$$

برخی اندازه‌گیری و روابط دیگر (نظیر دراز، پهنا، مدار متوسط، وضوح و... جدول اول) بطور دقیقی تعریف شده‌اند (برای کسب اطلاعات بیشتر به مقالها Ashish ghosh, Sankar k.pal و Azriel Resenfeld مراجعه نمائید). کاربرد هندسه فازی در بپسود تصویر، بخشنده تصویر و نمایش تصویر می‌باشد.

موضوعات مختلف هندسه فازی در پردازش تصویر :

موضوع:	قبل‌آمدهای فازی بسط داده شده
توبولوژی دیجیتالی	همبندی، محاط بودن، مجاورت، ...
متريک	مساحت، محیط، قطر، ...
اندازه‌های اشتاقاچی	فسردگی، فهرست، پوشش ناحیه، وضوح، کشیدگی، ...
فرم‌های اوليه	دواير، مثلث‌ها، مربعات، ...
محدب بودن	بسط دو بعدی، محدب و مقعر بودن مجموعه‌های فازی
اسكلت	مدار متوسط، نازکی، اسكلتبندی

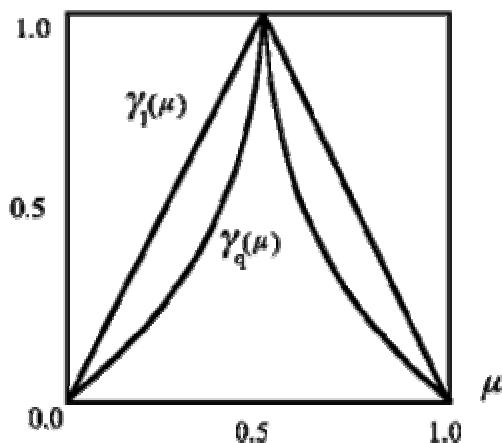
اگر ما یک تصویر (یا بخش‌های آنرا) به عنوان زیر مجموعه‌های فازی بنگریم، باید قادر باشیم راهی برای تعیین میزان فازی بودن یک تصویر بیابیم. افزایش یا کاهش میزان فازیت یک تصویر در برخی مسائل پردازش تصویر مانند بهمود، بخش‌بندی می‌تواند به کار آید. Arnold Kaufman شاخص (خطی) فازیت را اینگونه بیان می‌کند.

$$\gamma_1 = \frac{2}{MN} \sum_m \sum_n \min(\mu_{mn}, 1 - \mu_{mn})$$

برای هنگامی که تصویری به اندازه  $M^*N$  داشته و می‌خواهیم فازیت را با توجه به تفاوت ما بین مقادیر عضویت و مکمل‌های آنها بیابیم، شاخص مربعی فازیت به طریق مشابه قابل تعریف است:

$$\gamma_q = \frac{2}{\sqrt{MN}} \left[ \sum_m \sum_n \left\{ \min(\mu_{mn}, 1 - \mu_{mn}) \right\}^2 \right]^{\frac{1}{2}}.$$

میزان فازیت زمانی که همه عضویت‌ها صفر یا یک باشند برابر صفر است (مجموعه معمولی: تصویر باینری) فازیت هنگامی که مقدار عضویت‌ها ۰/۵ باشند به ماقریم مقدار خود می‌رسد.



قطع‌آ روش‌های دیگری برای تعیین میزان فازیت یک تصویر ارائه شده است. Termin و Deluca روش لگاریتمی را ارائه کردند:

$$H_{\log}(X) = \frac{1}{MN \ln 2} \sum_m \sum_n S_n(\mu_{mn}),$$

که

$$S_n(\mu_{mn}) = -\mu_{mn} \ln \mu_{mn} - (1 - \mu_{mn}) \ln(1 - \mu_{mn}).$$

در کنار این تعاریف، روش‌های دیگری نیز برای اندازه‌گیری اطلاعات تصویری که بصورت مجموعه‌های فازی تعریف شده‌اند وجود دارند، برای مثال:

Fuzzy correlation

Fuzzy expected value

Weighted fuzzy expected value

Fuzzy divergence

Hybrid entropy

...

## تشخیص لبه فازی

راههای مختلفی برای افزایش دقت در تشخیص لبه فازی وجود دارد:

- تعریف تابع عضویت مناسب:

این آسانترین راه است. این روش با تعریف تابع عضویت مناسب که نشانگر درجه لبه‌گی در هر همسایگی است؛ انجام می‌شود.

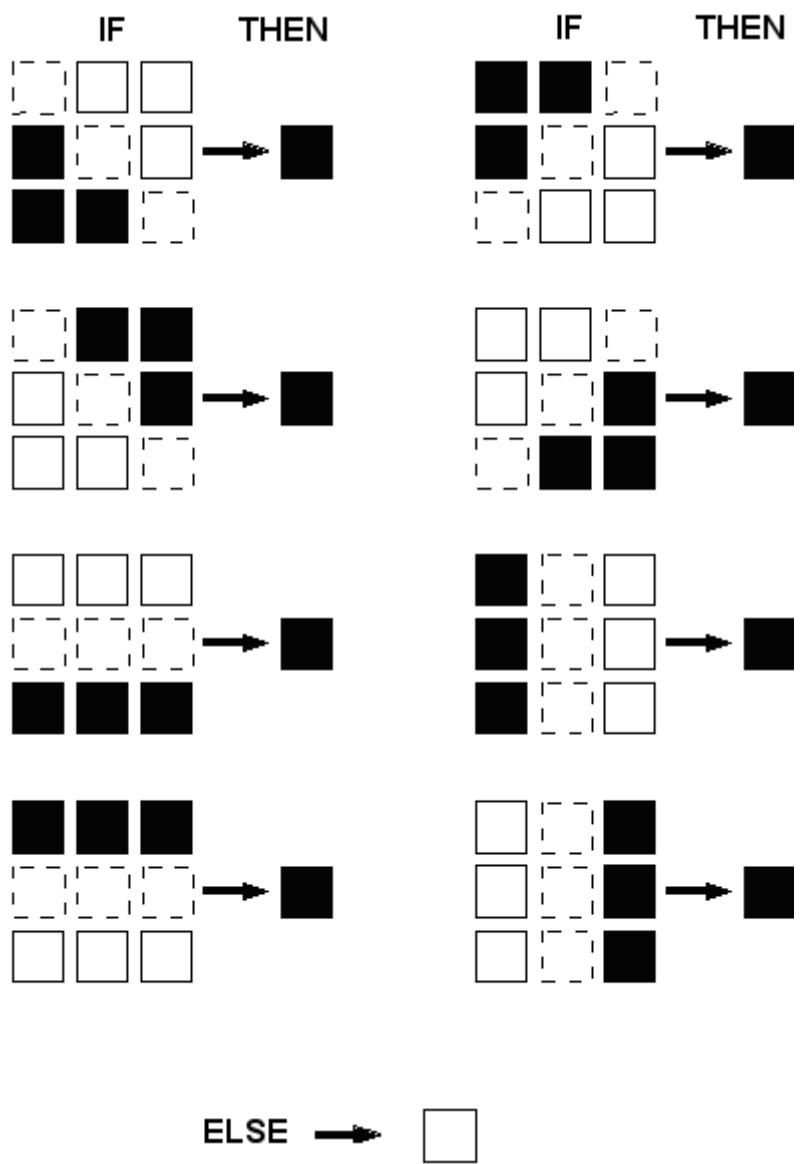
$$\mu_{\text{Edge}}(g(x,y)) = 1 - \frac{1}{1 + \frac{\sum_N |g(x,y) - g(i,j)|}{\Delta}}$$



این رهیافت تنها زمانی می‌تواند درست باشد که مفاهیم فازی به منظور اصلاح ارزش‌های عضویت به کار رود. تابع عضویت به صورت تعیین شده، این روش سریع است ولی کارایی محدودی دارد.

- تشخیص لبه فازی قانون‌گرا:

استفاده از قوانین if – then – قابلیت گسترش کارایی لبه‌یاب‌های عمومی یا ویژه را در همسایگی از پیش تعریف شده دارند:



ELSE →



1- در سال‌های اخیر بسیاری از محققان تئوری‌های مجموعه فازی را برای یافتن تکنیک‌های نوین توسعه کنتراست بکار گرفته‌اند و به رهیافت‌های جالبی هم رسیده‌اند که برخی از آن به شرح زیر است:

1-1- توسعه کنتراست به وسیله عملگر INT (Pal/King 1981/1983)

گام‌ها:

گام 1- تعریفتابع عضویت

$$\mu_{mn} = G(g_{mn}) = \left[ 1 + \frac{g_{max} - g_{mn}}{F_d} \right]^{-\frac{1}{F_t}}$$

گام 2- اصلاح ارزش‌های عضویت

$$\mu'_{mn} = \begin{cases} 2 \cdot [\mu_{mn}]^2 & 0 \leq \mu_{mn} \leq 0.5 \\ 1 - 2 \cdot [1 - \mu_{mn}]^2 & 0.5 \leq \mu_{mn} \leq 1 \end{cases}$$

گام 3- ساخت لایه‌های خاکستری جدید.

$$g'_{mn} = G^{-1}(\mu'_{mn}) = g_{max} - F_d \left( (\mu'_{mn})^{\frac{1}{F_t}} - 1 \right)$$

1-2- توسعه کنتراست با استفاده از (Craig and Schneider 1992) Fuzzy Expected value

گام‌ها:

گام 1- محاسبه هیستوگرام تصویر

گام 2- تعیین Fuzzy Expected value

گام 3- محاسبه فاصله لایه‌های gray-level از FEV

$$D_{mn} = \sqrt{(FEV)^2 - (g_{mn})^2}$$

گام 4- ساخت لایه‌های خاکستری جدید.

$$g'_{mn} = \max(0, FEV - D_{mn}) \quad \text{if } g_{mn} < FEV$$

$$g'_{mn} = \min(L-1, FEV + D_{mn}) \quad \text{if } g_{mn} > FEV$$

$$g'_{mn} = FEV \quad \text{otherwise.}$$

1-3- توسعه کنتراست به وسیله اغراق هیستوگرام فازی (Tizhoosh 1995/1997)

گام‌ها:

گام 1- تنظیم شکل تابع عضویت (با توجه به تصویر واقعی)

گام 2- تنظیم مقدار B فازی‌گر (یک linguistic hedge)

گام 3- محاسبه مقادیر عضویت

گام 4- اصلاح ارزش‌های عضویت به کمک linguistic hedge

گام 5- ساخت لایه‌های خاکستری جدید

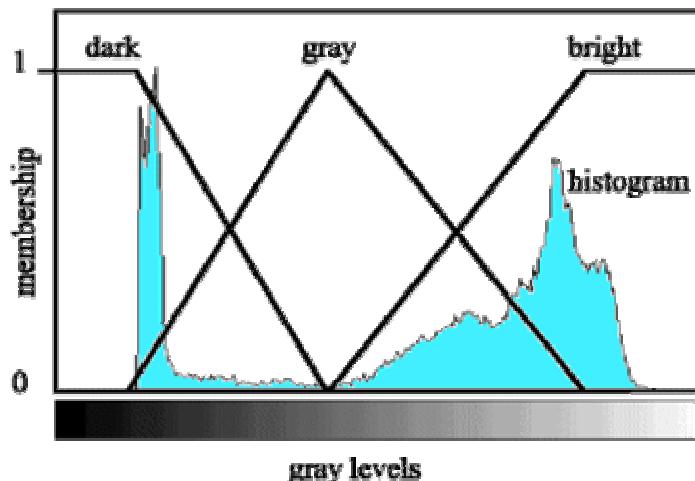
$$g'_{mn} = \left( \frac{L-1}{e^{-1}-1} \right) \cdot \left[ e^{-\alpha_{mn}(g_{mn})^{\beta}} - 1 \right]$$

1- توسعه کنتراست بر پایه قوانین (Tizhoosh 1997) If- Then

گام ها :

گام 1- تنظیم پارامترهای سیستم استنباط (ویژگی های ورودی، توابع عضویت)

گام 2- فازی سازی پیکسل های واقعی (عضویت دهی به مجموعه های تاریک، خاکستری و روشن پیکسل ها)



گام 3- استنباط (برای مثال اگر تاریک باشد، تاریکتر شود، اگر خاکستری، خاکستری. اگر روشن، روشنتر).

گام 4- از حالت فازی خارج کردن نتایج استنباط با استفاده از سه برج برنده

1- بهبود کنتراست Locally Adaptive

در حالت های بسیاری، تکنیک های سراسری نتایج رضایت بخشی را در اختیار قرار نمی دهد بنابراین و برای به دست آوردن نتایج بهتر لازم است از پیاده سازی روش های محلی استفاده کنیم.

2- تیز کردن تصویر و کاهش نویز

با این تفاسیر، ما قادر خواهیم بود تکنیک های جدیدی را برای تیز کردن تصویر و کاهش پارازیت های آن بیابیم.  
نمونه هایی چند آمده است:

•

A Fuzzy Filter for Images Corrupted By Impulse Noise •

Russo, F., Ramponi, G., SPLetters(3), No. 6, June 1996, pp. 168-170

A design method of fuzzy weighted median filters •

Taguchi, A. [Akira], ICIP96(16A9), 1996

Weighted fuzzy mean filter for image processing •

Lee, C.-S., Kuo Y.-H., Yau, P.-T., Fuzzy Sets and Systems, vol.89, no.2, pp. 157-180, 1997

A robust approach to image enhancement based on fuzzy logic •

Choi, Y.S., Krishnapuram, R., IEEE Trans. on Image Processing, vol. 6, No. 6, pp. 808-825, 1997

## بخش‌بندی تصاویر فازی:

اجزاء تئوریک مختلف از پردازش تصویر فازی امکانات مختلفی از تکنیک‌های جدید بخش‌بندی تصویر در اختیار قرار می‌دهد. جدول زیر توضیح مختصری از تکنیک‌های فازی در پردازش تصویر را در اختیار قرار می‌دهد.

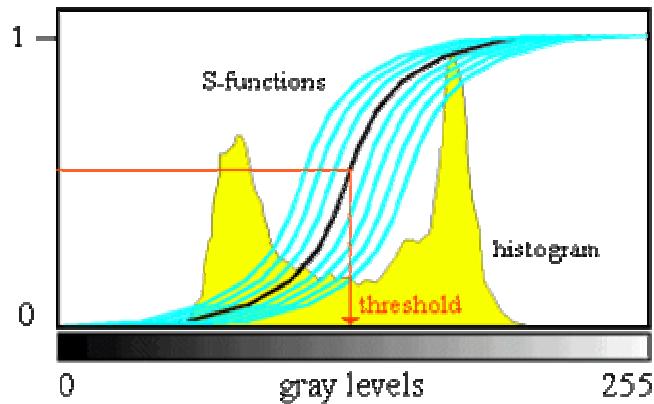
مهمترین رهیافت‌های فازی در بخش‌بندی تصاویر:

رهیافت فازی	توضیح مختصر
الگوریتم‌های کلاسیکهای فازی	کلاسیکهای فازی یکی از قدیمی‌ترین رهیافت‌ها به پردازش فازی می‌باشد. الگوریتم‌های نظری C-means (Bezdek, FVM) و احتمالی Kishnaparam (S Keller, PCM) که می‌تواند به عنوان شباهت و سازگاری با یک شی ایده‌آل یا یک ویژگی معین تفسیر شود.
رهیافت قانون‌گرای فازی	اگر ما ویژگی‌های تصویر را به عنوان متغیرهای زبانی تفسیر کنیم، به راحتی قادر خواهیم بود با تعریف قوانین if-then تصویر را به نواحی مختلف بخش‌بندی کنیم. یک قانون بخشندی ساده عبارتست از: اگر پیکسل تاریک است و همسایگی آن هم تاریک است و یکنواخت است پس متعلق به پس‌زمینه است.
انتگرال فازی	انتگرال فازی به فرم‌ها مختلف قابل استفاده است: بخش‌بندی با استفاده از وزن دهی به ویژگی‌ها (اندازه‌های فازی بیانگر اهمیت ویژگی‌های معینی خواهد بود). ترکیب نتایج از الگوریتم‌های بخش‌بندی مختلف (استفاده بهینه از مزیت‌های انفرادی). بخش‌بندی به وسیله ترکیب سنسورهای مختلف (استفاده بهینه از مزیت‌های انفرادی) بخش‌بندی به وسیله ترکیب سنسورهای مختلف (برای مثال تصاویر، چند طیفی؛ اندازه‌های فازی بیانگر ارتباط و اهمیت هر سنسور خواهند بود).
اندازه‌گیری فازیت و اطلاعات تصویر	اندازه‌گیری فازیت (برای مثال آشفتگی فازی) و اطلاعات تصویر (برای مثال دیوار ژانس فازی) نیز می‌تواند در بخش‌بندی و آستانه‌گیری به کار بسته شود.
هندرسه فازی	اندازه‌های هندسی فازی مانند فشردگی فازی و شاخص پوشش ناحیه نیز می‌تواند برای اندازه‌گیری فازیت هندسی نواحی مختلف یک تصویر استفاده شود. بهینه‌سازی این اندازه‌ها (برای مثال کمینه‌سازی فشردگی فازی با توجه به نقطه متقاطع از تابع عضویت) می‌تواند برای طبقه‌بندی‌های پیکسلها بکار آید.

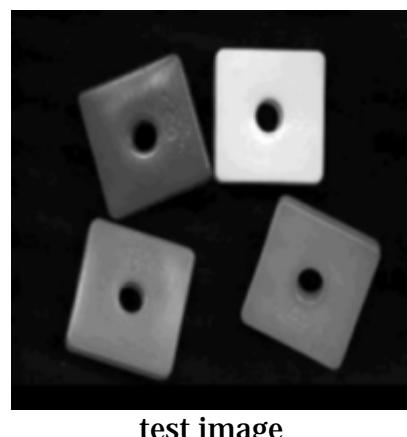
مثال: آستانه‌گیری به وسیله کمینه‌سازی فازیت

تکنیک‌های متعدد کلاسیکی برای آستانه‌گیری وجود دارند. در سالهای اخیر، برخی نویسنده‌گان با استفاده از ویژگی فازیت تصویر روش‌های جدیدی ارائه نموده‌اند (مقالاتی Pal/Murthy 1990, Rosenfeld/Pal 1988, Huang/Wang 1995). برای مثال:

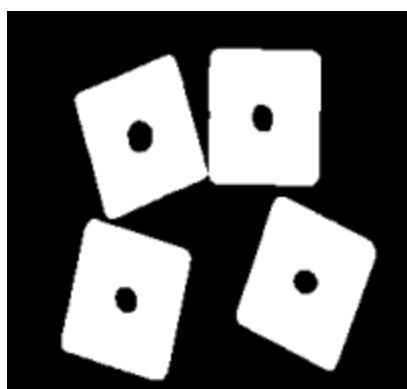
یک تابع عضویت استاندارد  $S$ , پیکسل به پیکسل روی محدوده سطوح خاکستری حرکت داده شده است: (شکل اول): در هر موقعیت میزان فازیت محاسبه شده است؛ موقعیتی که کمترین مقدار فازیت را دارد آستانه مناسبی است. شکل دوم که مقایسه‌ای است بین این روش و الگوریتم آستانه‌گیری otsa.



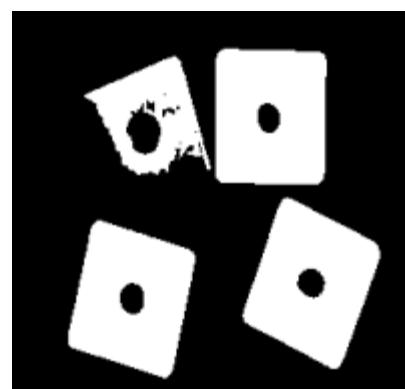
یافتن مینیمم فازیت به عنوان ابزاری برای آستانه‌گیری بخش‌ها



test image

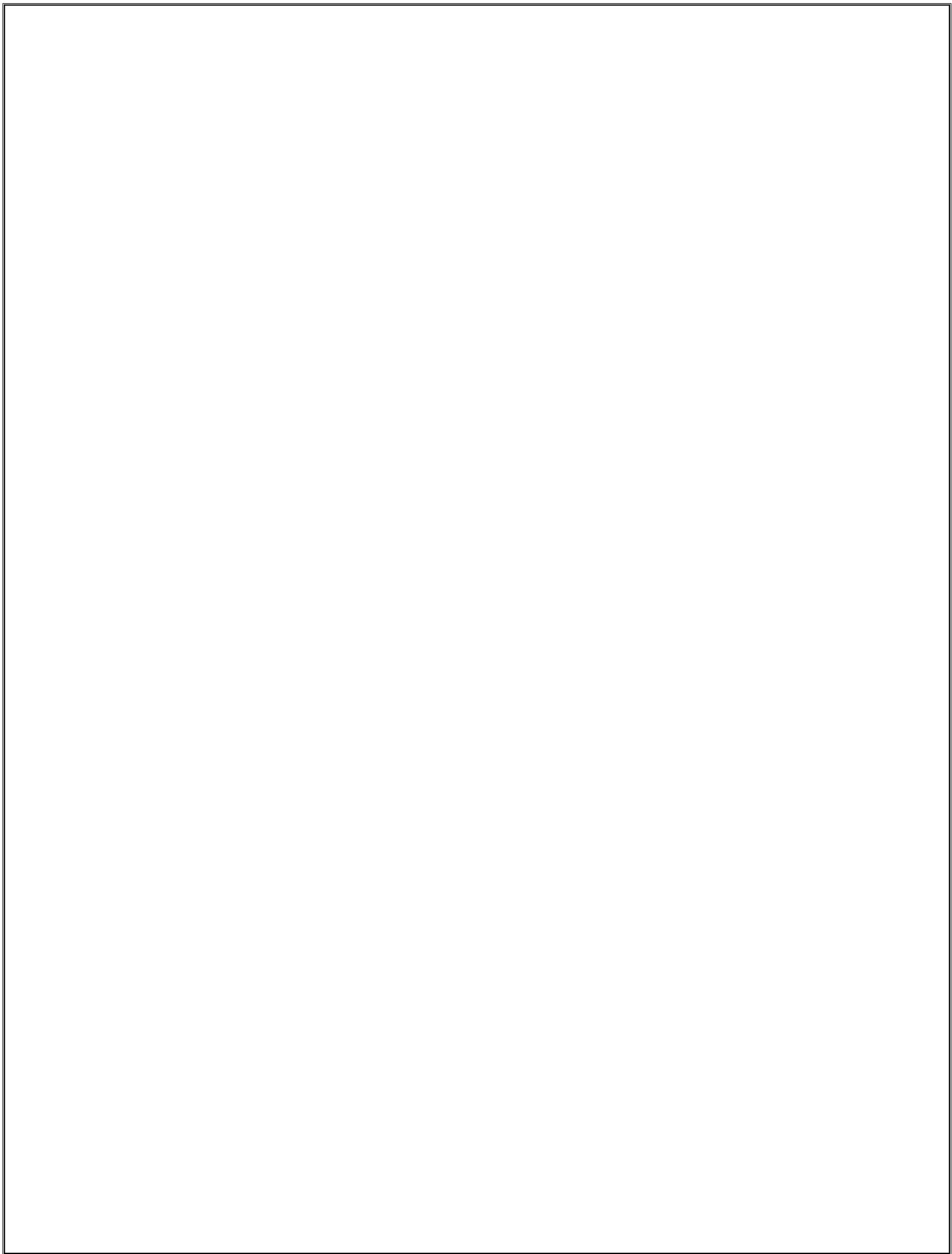


thresholded by fuzzy method



thresholded by Otsu algorithm

مثالی برای آستانه‌گیری فازی به وسیله کمینه‌سازی فازیت تصویر



پیوست

# Image Processing in MATLAB

## ◆ آشنایی با Toolbox Image Processing و Matlab

Toolbox از امکانات جنبی برنامه Matlab است. برای اینکه مطمئن شوید که این Toolbox روی Matlab شما نصب شده است؛ دستور ver را اجرا کنید. این دستور لیست هرچه که از Matlab روی رایانه شما نصب شده است را ارائه می‌دهد. بین Matlab 5.x و Matlab 6 برای کار پردازش تصویر تفاوت چندانی وجود ندارد اما مثل همیشه نسخه جدیدتر امکانات بیشتری دارد که البته فعلاً با آنها کاری نداریم. دستورهای معرفی شده: ver

## ◆ چگونه یک فایل تصویر را در Matlab باز کنیم

Matlab می‌تواند فایل‌های گرافیکی با فرمتهای JPEG, TIFF, GIF, BMP, PNG, HDF, PCX, XWD, ICO, CUR را به عنوان فایل گرافیکی بخواند. مثلاً برای وارد کردن تصویری به نام cameraman.tif به فضای Matlab کافی است از دستور imread استفاده کنیم:

```
MyImage=imread('cameraman.tif','tif');
```

توجه داشته باشید که دستور خواندنش را می‌دهید باید برای برنامه قابل دسترس باشد. یعنی یا باید در مسیر (Path) Matlab باشد یا اینکه در پروندهای (folder) قرار داشته باشد که در حال حاضر برنامه به آن دسترسی دارد. برای اینکه بدانید که Matlab برای پیدا کردن فایلی که دستورش را دادید کجا را خواهد گشت اینکارها را بکنید: از دستور path برای اینکه بدانید کدام پروندهای جزء مسیر پیش‌فرض Matlab است و از دستور dir برای اینکه بدانید که Current Directory چیست؛ استفاده کنید.

خب تا اینجا یک فایل تصویر را در محیط Matlab وارد کرده‌ایم. همانطور که می‌دانیم یک تصویر دیجیتال بر روی کامپیوتر در قالب یک ماتریس ذخیره می‌شود. پس MyImage یک ماتریس است. برای اینکه بدانیم فایل خوانده شده از چه فرمتی است(سیاه سفید، یا Gray Scale یا رنگی ) می‌نویسیم:

```
imfinfo('cameraman.tif')
```

این دستور را اجرا کنید و ببینید چه می‌نویسد... اما اگر بخواهید بدانید که ماتریس ذخیره شده MyImage از چه نوعی است کافی است بنویسد: whos و لیست متغیرهای مقیم شده در حافظه و نوع و اندازه آنها را ببینید. دستورهای معرفی شده: imread, imfinfo, whos, path, dir

## ◆ چطور تصویر را ببینیم؟

خب حالا می‌خواهیم تصویر را که در یک ماتریس ذخیره شده است را ببینیم. بنویسید: (MyImage)imshow

جالب است نه؟ فکر می‌کنید اگر بخواهیم دو تا تصویر را با هم ببینیم باید چکار کنیم؟ این را امتحان کنید:

```
YourImage=imread('tire.tif','tif');
```

```
figure
```

```
subplot(1,2,1), imshow(MyImage), title('MyImage')
```

```
subplot(1,2,2), imshow(YourImage), title('YourImage')
```

با اجرای این دستورات به آن چیزی که اتفاق افتاد توجه کنید؛ حتماً متوجه می‌شود که هر کدام از این دستورات چکار می‌کنند.

دستورهای معرفی شده: imshow, subplot, title, figure

## ❖ تصویر را خواندیم حالا چکار کنیم؟

کمی نویز دستوپا می کنیم و به تصویر اضافه می کنیم که بعداً راهی پیدا کنیم حذفش کنیم:  
;('pepper & MyImage,'salt);imnoise=imagen  
(imagen)imshow

دستور **imnoise** نویزهای مختلفی را در اختیار ما می گذارد که به تصویر اضافه کنیم. افزودن نویز برای شبیه سازی اشکالاتی است که ممکن است به هر سیستم پردازش تصویر وارد شود. اینجا فرض کردہایم که نویز «نمک و فلفل» به تصویر اضافه شده است! اسمش عجیب غریب است؟ این نویز را روی تصویر تلویزیون تان اگر آتن درست تنظیم نباشد حتماً دیدهاید. می دانیم که وقتی نویز داریم با یک فیلتر حذفش می کنیم. فیلتری که انتخاب می کنیم باید مناسب نویزی باشد که روی تصویر سوار شده است. بهترین فیلتر برای نویز «نمک و فلفل» فیلتر میانه است که در **Medfilt2** با دستور **Matlab** قابل استفاده است:

```
figure  
imagefilt=medfilt2(imagen);  
imshow(imagefilt)
```

برای دست گرمی هم که شده سعی کنید تصویر نویز و تصویر فیلتر شده را در یک صفحه واحد نمایش دهید.  
دوباره به تصویر **MyImage** نگاه کنید. فرض کنید می خواهیم مارک دوربین درون عکس را از روی شکل آن تشخیص دهیم. بازهم فرض کنید به روشی که بعداً بیشتر راجع بهش صحبت می کنیم فهمیدیم که دوربین درون این مختصات از تصویر قرار دارد. بین سطر 59 تا 84 و ستون 131 تا 170. پس از همین مختصات تصویر را با دستور **imcrop** می بریم.

```
imagecrop=Imcrop(MyImage,[ 131 59 39 25]);
```

در این دستور مختصات برش را اینطور می نویسم [ارتفاع، پهنای، حداقل Y، حداقل X]. این قسمت بریده شده را نمایش بدهید و ببینید. در مرحله بعد سیستم تشخیص دهنده نوع دوربین احتیاج دارد که اندازه تصویر مقدار خاصی مثلاً 120x100 باشد. پس باید اندازه تصویر برش داده شده را تغییر دهیم:

```
imagesz=imresize(imagecrop,[120 100]);
```

تصویر جدید را نگاه کنید. می بینید که به اندازه جدید در آمده است.

دستورهای معرفی شده: **imnoise, medfilt2, imcrop, imresize**

## ❖ کمی خلاقیت مهندسی!

یک فایل جدید باز می کنیم که عکسی از دانه های برنج است. می خواهیم دانه های برنج را درون عکس بشماریم. این عکس از قسمت برنج مرغوب خط تولید یک کارخانه است. شمارش دانه های برنج به کارخانه برنج سازی کمک می کند بداند چند درصد برنج هاییش درجه یک هستند. این دستورات را اجرا کنید:

```
p=imread('rice.tif');  
re=edge(p,'sobel');  
imshow(re)
```

```
re2=edge(p,'canny');  
figure, imshow(re2)
```

البته اول تصویر اصلی برنج ها را ببینید و سپس مقایسه کنید. دو روش **Canny** و **Sobel** برای پیدا کردن لبه ها مورد آزمایش قرار گرفته است. مشخص است که روش **Canny** اینجا بهتر جواب داده است. دو کار به عهده شماست.

a. الگوریتمی پیدا کنید که تعداد برنج‌های درون عکس را بشمارد.(راهنمایی: اگر بتوانید تعداد مسیرهای بسته‌ی درون تصویر لبه‌ها را بشمارید تقریباً همه برنج‌ها را شمرده‌اید)

b. کشف کنید که روش Canny برای مشخص کردن لبه چه روشی است و چطور عمل می‌کند و چرا از روش ساده Sobel بهتر است.

برای بهتر شده نتیجه کار با روش Sobel بهتر است از عملگرهای ریخت‌شناسی (morphological) استفاده کنیم. سعی می‌کنیم قسمت‌های نزدیک به هم لبۀ هر برنج را به هم متصل کنیم. با این کار لبۀ برنج‌ها را کامل می‌کنیم و خوردگی‌ها را از بین می‌بریم. اما قبل از ادامه کار دستور close all را امتحان کنید. این دستور همه figure هایی که باز هستند را می‌بندد تا زیاد شلوغ نشود! حالا این دستورها را اجرا کنید:

```
figure, imshow(re)
re4=bwmorph(re,'close');
re5=bwmorph(re4,'bridge');
figure, imshow(re5), title('after morphological operations')
```

آخرین تصویر لبه‌ها را با تصویر اولیه لبه‌ها که همان re است مقایسه کنید. می‌بینید که در متصل کردن قطعات پیشرفت خوبی داشتیم. اولین دستور Closing عمل bwmorph را روی تصویر انجام می‌دهد و دومین دستور عمل Bridge. برای اینکه بدانید هر کدام از این دستورات چکار می‌کنند از help bwmorph استفاده کنید.

برای اینکه بهتر کار با این دستورها را یاد بگیریم سری به Matlab Demo می‌زنیم. در صفحه Matlab بنویسید: demo و با اجرا کردن این دستور لیست demo‌های مختلف ظاهر می‌شود. Image Processing Toolbox را انتخاب کنید. و در لیست کناری Edge Detection را انتخاب کنید. بقیه ماجرا با خودتان! دستورهای معرفی شده: edge, bwmorph, demo

#### ❖ هیستوگرام تصویر:

تا به حال تصویرهای Gray Scale را بررسی کردیم. از این به بعد هم همین کار را می‌کنیم! با این تفاوت که اگر تصویر ورودی رنگی بود اول Gray Scale می‌کنیم و بعد کارها را ادامه می‌دهیم.

تصویر flowers.tif را با دستور imread در ماتریس flowers ذخیره کنید. از دستور rgb2gray برای تبدیل فرمت رنگی تصویر به Gray Scale استفاده کنید. حالا هیستوگرام تصویر را رسم کنید. بعد یکی از قله‌های نمودار هیستوگرام را انتخاب کنید که بین دو تا دره باشد. کاری کنید که فقط این قله درون هیستوگرام باقی بماند. بعد با دستور im2bw تصویر را از Scale به سیاه سفید تبدیل کنید و نتایج را مقایسه کنید! چطور شد؟ توضیح می‌دهم، اینطوری:

```
flowers=imread('flowers.tif','tif');
flowersgray=rgb2gray(flowers);
imhist(flowersgray)
```

من هیستوگرام را نگاه کردم و شما هم نگاه کنید. (هیستوگرام تصویر را با دستور imhist کشیدم) قله بین 50 و 100 را انتخاب می‌کنم. حالا یک ماتریس جدید می‌سازم طوری که فقط جاهایی که مقدار ماتریس flowersgray در محدوده 50 تا 100 است، در ماتریس جدید 0 باشد و باقی جاهای 1 باشد. (یعنی یک ماتریس باینری یا سیاه سفید):

```
f=ones(size(flowersgray));
f(find(flowersgray>50&flowersgray<100))=0;
imshow(f)
```

دستورهای بالا به این معنی است: ماتریس f را هماندازه ماتریس flowersgray بساز. همه درایه‌هایش را یک بگذار. بعد دستور find درون ماتریس flowersgray جستجو می‌کند و شماره درایه‌هایی از ماتریس flowersgray را به من می‌دهد که در شرط

روبهرو برایش مشخص شده. (همان محدوده 50 تا 100). این درایه‌های بخصوص را در ماتریس  $f$  از 1 به 0 تبدیل می‌شود. نتیجه را باهم ببینیم!

اگر دقت کنیم می‌بینیم که مجموعه گلدون از پس زمینه جدا شد. آنهم تنها با انتخاب یک قله درون هیستوگرام... حالا من از کجا فهمیدم که این قله مهم است؟... لابد علم غیب داشتم؟... بعدا راجع به انتخاب قله‌ها بیشتر خواهیم فهمید.

حالا دستور `im2bw` را - که تصویر را سیاه سفید می‌کند - امتحان کنید.

```
bwflowers=im2bw(flowersgray,gray);  
imshow(bwflowers)
```

دستور `im2bw` حدی تعیین می‌کند و درایه‌هایی که از آن حد کمتر باشند را صفر و آنهایی که بیشتر هستند را یک می‌کند. آیا متوجه تفاوت دو روش شده‌اید؟ اگر به جای گلدون در تصویر مورد نظر، عکس پرسنلی افراد بود چه اشکالی در روش `im2bw` وجود داشت؟

دستورات معرفی شده: `rgb2gray`, `imhist`, `ones`, `find`

#### ◆ اگر ما روبات بودیم...

خب فرض کنید برنامه‌ای برای یک روبات می‌نویسید که با استفاده از یک دوربین به یک بازوی دقیق فرمان می‌دهد که سکه‌های «اسکروچ» را بردارد و روی هم قرار دهد. برای اینکار لازم است محل دقیق سکه‌ها را تشخیص دهیم تا فرمان مناسب برای بازو صادر شود. برنامه زیر این کار را انجام می‌دهد. دقت کنید ببینید که چطور اینکار را انجام می‌دهد:

```
coins = imread('eight.tif');  
BW=~im2bw(coins,.75);  
imshow(BW)  
L = bwlabel(BW);  
stats = imfeature(L,'Centroid');  
stats(7)
```

خروجی همان متغیر `stats` است. این متغیر 7 عنصر دارد. درحالی که 4 سکه بیشتر درون تصویر وجود ندارد! اشکال از کجاست؟.... راه حلی هم برای این به ذهنتان می‌رسد؟

دستورهای معرفی شده: `bwlabel`, `imfeature`

## RECOGNITION OF PERSIAN ONLINE HANDWRITING USING ELASTIC FUZZY PATTERN RECOGNITION

RAMIN HALAVATI\* and SAEED BAGHERI SHOURAKI†

*Sharif University of Technology, Tehran, Iran*

\*halavati@ce.sharif.edu

†sbagheri@ce.sharif.edu

Persian is a fully cursive handwriting in which each character may take different forms in different parts of the word, characters overlap and there is a wide range of possible styles. These complexities make automatic recognition of Persian a very hard task. This paper presents a novel approach on recognition of such writings systems which is based on the description of input stream by a sequence of fuzzy linguistic terms; representation of character patterns with the same descriptive language; and comparison of inputs with character patterns using a novel elastic pattern matching approach. As there is no general benchmark for recognition of Persian handwriting, the approach has been tested on the set of words in first primary Iranian school books including 1250 words resulting in 78% correct recognition without dictionary and 96% with dictionary.

*Keywords:* Online handwriting recognition; elastic pattern matching; fuzzy modeling.

### 1. Introduction

Persian character set and writing style along with other members of Arabic family character set are used by more than 30% of the world's population and serve in the writing of many widespread languages such as Farsi, Arabic, and Urdu.<sup>2</sup> In contrast to advances in online Latin and Far East handwriting recognition, relatively few studies have been devoted to these languages. This is mainly due to its highly cursive nature which makes it a very complex pattern for both segmentation and recognition: characters are written in up to four different forms in different parts of word, they usually stick together, they may overlap, and there are several different writing styles (Figs. 1–3).

A number of online handwriting recognition systems have been proposed in the last two decades incorporating stochastic, neural network, model matching or structural/syntactical techniques such as Refs. 1, 4–8, 10, 11, 14, 15, 19 and 20. A detailed review of these approaches is presented in Sec. 6 during comparison of the presented approach with them. The major problem of all existing methods is high sensitivity to writing perturbations due to the vast variety of writing styles which results in imprecise and slow recognition or writer dependency.

Letter Name:	Alef	Pe	Jhe	Shin	Ein	He
<b>Isolated Form:</b>	ا	پ	ژ	ش	ع	ه
<b>Beginning of the Word:</b>	ا_س_ب	پ_ا	ژ_ر_ف	ش_ب	ع_ق_ر_ب	ه_ر_ا_ت
<b>Middle of the Word:</b>	ب_ار_ان	ت_ي_ه	م_ڑ_ه	ن_ش_ان	ب_ع_د	م_ه_ي_ا_ر
<b>End of the Word:</b>	دا_را	ل_پ	ت_ا_ز	م_ي_ش	ب_ر_ق_ع	ر_ف_ت_ه

Fig. 1. Different forms of a character based on its position.

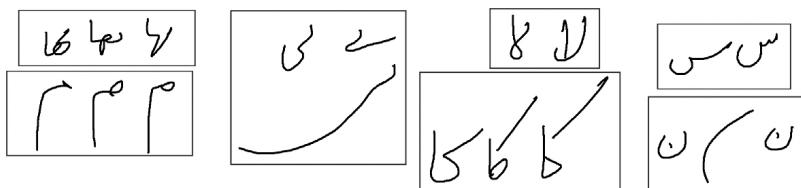


Fig. 2. Different writing variation samples by different people. All writings within a rectangle are the same characters, written by different people.



Fig. 3. A sample of Persian calligraphy. Most of the writing rules can be neglected for the sake of beauty and harmony. This paper deals only with normal writing styles.

Considering the limitations and drawbacks of the existing systems, it can be concluded that a practical handwriting recognition method for complex writing systems such as Persian should be both fast and flexible. To achieve these requirements, the knowledge base must be small but robust and changes in style or orientation should be covered by flexible prototypes that contain widely valid description of character information.<sup>16</sup>

To have the above features together, this paper presents a fuzzy elastic approach with the following three major characteristics: first, it represents input data and patterns using a fuzzy linguistic description that is derived from a case study on school kids that learn how to read and write Persian. This description method results in ignorance of some perturbations and more robustness when dealing with different writing styles. Second, the proposed recognition approach performs the recognition and segmentation of input stream simultaneously and therefore overcomes the common problem of recognition/segmentation priority. And third, the recognition part uses a novel elastic approach that is very much robust to writing innovations and perturbations.

The rest of paper is organized as follows: In the next section, we discuss the main rationales behind our method and the general ideas. Section 3 represents the segmentation and description processes. In Sec. 4, the main pattern comparison algorithm is introduced and Sec. 5 presents our experimental results. Section 6 will be on results and feature's comparison between our approach and some other contributions and finally the conclusions and future works.

## 2. The Rationales Behind Our Method

Based on a case study on school kids at the very first stages of learning how to read and write Persian in Iranian schools, it was noted that they try to describe characters using simpler tokens called “lohe” which mainly consist of straight lines, arcs and circles. Separation of a written word into lohes by different children is not exactly the same and has a lot of variance but description of letters and words using these lohes is very much the same.<sup>12</sup>

These findings suggest that when a kid has not yet learned to recognize the word as a whole, s/he first tries to separate it into a set of linguistically describable terms and compare it with some other linguistic patterns which define letters and words. And as another important fact, this separation is not unique, but it is somehow compared with word/letter descriptions that the differences are ignored and the main pattern is recognized. Therefore, a description of input data and desired patterns by a set of fuzzy linguistic terms and a robust comparison between these terms can be an appropriate method to solve the hard problem of cursive handwriting recognition for computers, which are still in the primary school days of learning to solve human level problems.

To implement the above idea, we first try to separate the sequence of input points received from a pointing device into a sequence of tokens, each in the form of a line or an arc. To make the separation, we implement the fact that when the eye follows a sequence of points, it assumes a line, when the comprising vectors have almost similar angles, and it assumes an arc when succeeding vectors change their angles almost in a constant manner (Fig. 4). This idea is fully represented in Sec. 3.1.

Once the input is segmented, we can describe each segment with fuzzy linguistic terms by simple rules of thumb such as if its angle is around zero, it is a top to bottom part, etc. The characters are also described with similar linguistic terms and comprise a fuzzy knowledge base of patterns. When this is done, the input is described using expressions such as “*a short arc from top right to bottom left followed by a long line from bottom to top*” and it must match with a character pattern such as “*a small line from top to left, then a small line from left to right and finally a long line from bottom to left*”. The description method is expressed in detail in Sec. 3.2.

Due to unavoidable uncertainty in segmentation phase regarding writers' style, writing speed, writing device's friction, etc. the comparison between the input and

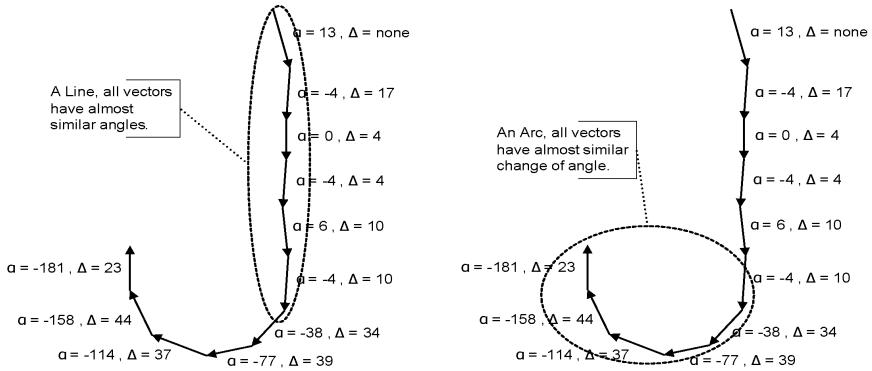


Fig. 4. Two samples for line and arc detection ideas. In both images, the angle of each vector ( $\alpha$ ) and its difference with previous vector's angle ( $\Delta$ ) are presented. In the left image, the selected vectors have almost similar angles (all around zero) and they compose a line and in the right image, selected vectors have similar change of angles (all around 35) and comprise an arc.

fuzzy patterns must deal with different unequal number of segments in input and pattern, unwanted segments, and differences between writer's style and defined patterns.

To cope with these necessities, we introduce a novel elastic fuzzy pattern comparison method which works as follows: at the reception of first segment of input, it generates some hypotheses on what the input is. By receiving further parts, it tries to produce new hypotheses based on the previous ones and prune those that are falsified. Once the input is complete, the hypothesis with the highest degree of belief is chosen as the matched pattern. This approach is described in detail in Sec. 4.

Different contributions exist for this task such as model matching approaches,<sup>13,22,24</sup> dynamic programming methods,<sup>9,17,23</sup> and dynamic time warping<sup>3</sup> mostly with the basic idea of a point to point mapping of inputs and patterns and a way of finding the best mapping. The major advantages of this approach compared to cited methods are, in general:

- (i) Description of input and pattern with fuzzy linguistic terms reduces the search space by a big factor.
- (ii) As the compared terms are all described with fuzzy terms, robustness versus perturbations is much more than point to point comparisons.
- (iii) The pruning methods prevent searching many unwanted states.

### 3. Segmentation and Representation Method

Based on the ideas presented in Sec. 2, this section presents the details on input segmentation, representation with fuzzy linguistic terms and character patterns definition.

### 3.1. Input primary segmentation

The input data is a sequence of vectors, separated from each other when the writer has picked up the writing device. We will call the sequence of points/vectors placing the pen on a writing device with its next pick up, an episode. Using this view, segmentation will be to separate an episode into one or more groups of consecutive vectors for which each composes one line or an arc. To do so and based on the ideas stated in Sec. 2, we start from the first point of an episode and compute the average of vector angles and average of vector angle changes (Formula 1,  $\text{AvgAngle}_i$  and  $\text{AvgDelta}_i$ ).

To have a measure of how much a point is a suitable candidate for being the end of a line or an arc,  $\text{LineMeasure}_i$  and  $\text{ArcMeasure}_i$  are computed as the absolute difference between vector's angle and angle change with the respective averages of previous points. Once both of these measures exceed certain thresholds, the point is marked as the end of the segment and the sequence restarts from the next point. We have practically chosen 7 and 20 as thresholds for  $\text{ArcMeasure}$  and  $\text{LineMeasure}$ , but as stated in Table 5 of the experimental results section, the system is not sensitive to these exact values. Once the segmentation is completed, we refine it by merging very small segments with their neighbor segments. Figure 5 presents the complete algorithm.

$$\begin{cases} \Delta_i = \text{Angle}_i - \text{Angle}_{i-1} & \text{LineMeasure}_i = |\text{Angle}_i - \text{AvgAngle}_{i-1}| \\ \text{AvgAngle}_i = \sum_{x=0}^i \text{Angle}_x / i & \text{ArcMeasure}_i = |\Delta_i - \text{AvgDelta}_{i-1}| \\ \text{AvgDelta}_i = \sum_{x=0}^i \Delta_x / i & \end{cases} \quad (1)$$

### 3.2. Input description

Once the input sequence is segmented, it is described using four fuzzy linguistic terms, namely segment's type, direction, length and direction of curvature:

#### 3.2.1. Segment type

Three descriptive terms are used for segment's type: line, arc and semicircle. To assign one of these terms to the segment, a straightness measure is used which was previously introduced in Ref. 24 with minor modifications. This measure is the ratio of the distance between the first and last points of the segment to the sum of the distances between consecutive points of the segment. This measure is presented in Eq. (2), and fuzzy membership sets that choose the appropriate linguistic term are presented in Fig. 6.

$$\text{Straightness} = 100 \times \exp \left( \frac{\text{dist}(P_n, P_0)}{\sum \text{dist}(P_i, P_{i+1})} \right) / (e - 1) \quad (2)$$

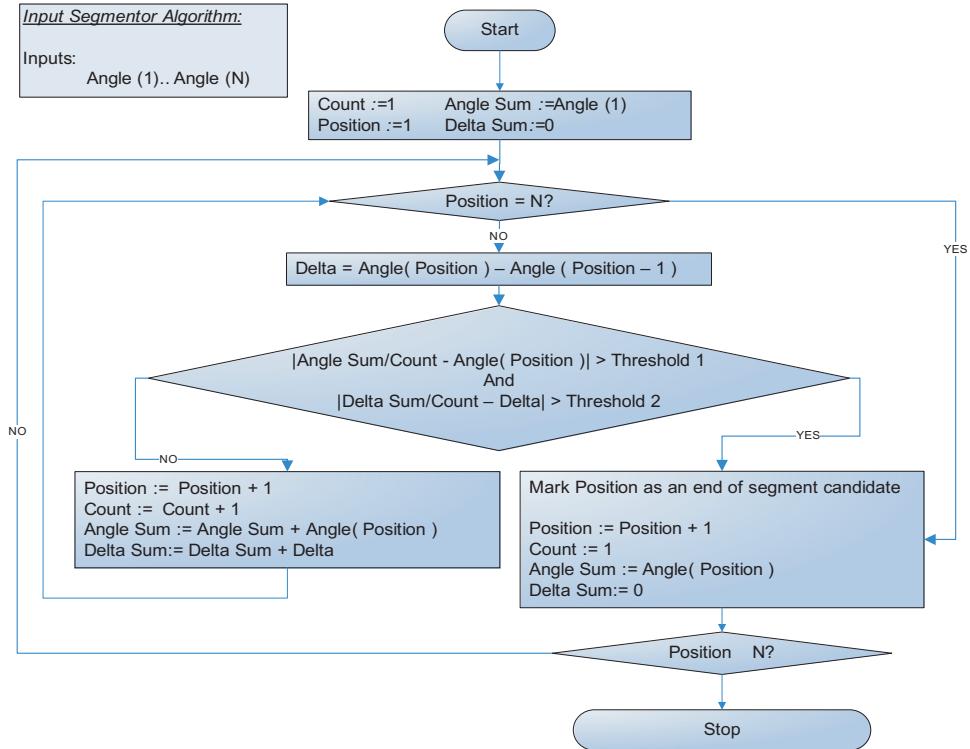


Fig. 5. The segmentation algorithm.

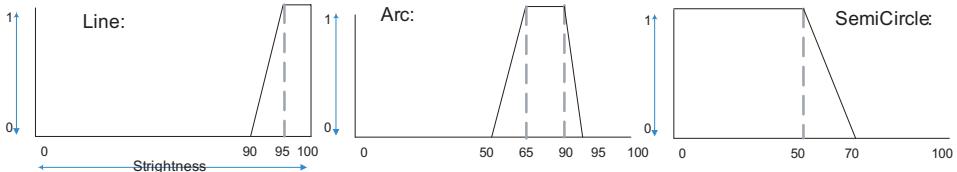


Fig. 6. Membership functions for segment type linguistic variables.

### 3.2.2. Segment direction

The second feature is the segment's direction which is described using eight fuzzy linguistic terms as shown in Fig. 7. The measure we have used for direction is the direct angle from the first point of the segment to its last point.

### 3.2.3. Curvature direction

If the segment is an arc or semi-circle, the next measure specifies the side of curvature on the direct line from the first point of the segment to its last point. To do so, the center of gravity of the segment is computed and the vector from the first

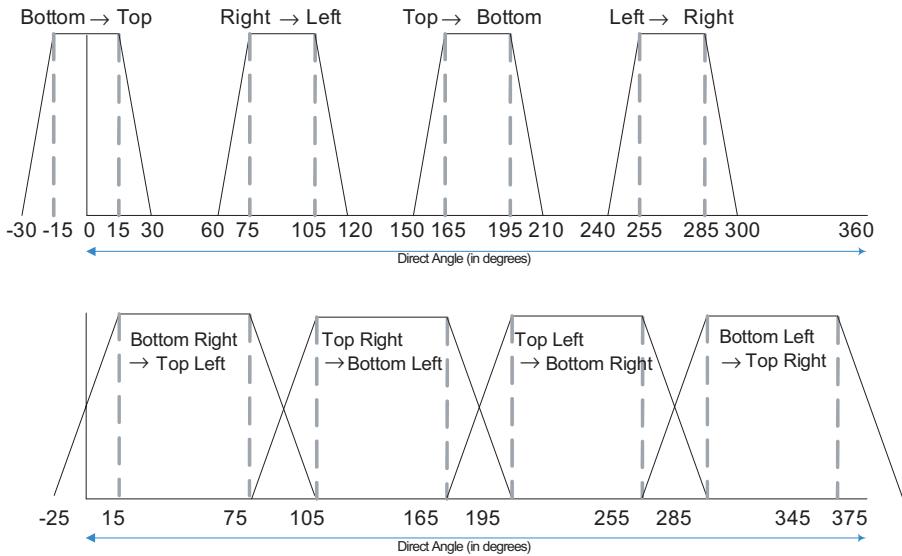


Fig. 7. Direction linguistic terms. Horizontal axis represents the direct angle from the first point of segment to its last point in degrees and vertical axis is the degree of belief in specified membership function.

point to COG is found. If this vector resides on the left of the direct vector from the first to last points, it is called clockwise curvature and otherwise, it is a counter clockwise curvature. This feature is the only non-fuzzy feature and has two crisp values: clockwise (C), counter clockwise (C.C). Figure 8 visualizes this measure.

### 3.2.4. Segment length

Segment length is the last and most trivial feature and is represented in Script Height Units (SHU) during computational phases and has five fuzzy linguistic terms in decision rule sets. SHU is a dynamic measure that is computed based on the

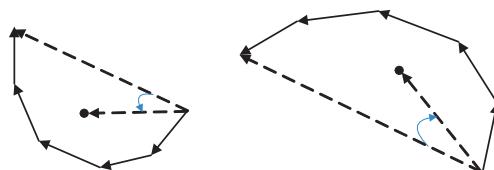


Fig. 8. Visualization of curvature direction. In the left picture, the point connecting the first point of the segment to its center of gravity (COG) lies on the left of the line connecting first to last points and is noted as Clockwise Curvature (C). In the right picture, the first point to COG lies on the right of the first to last points and therefore, it is called Counter Clockwise Curvature (C.C.).

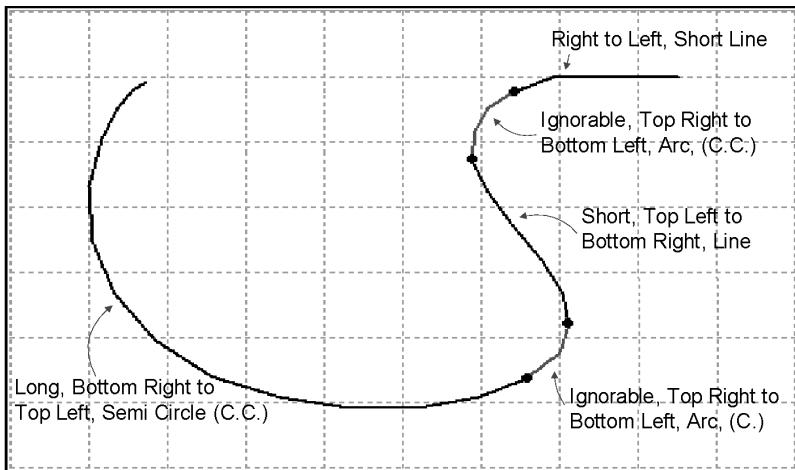


Fig. 9. A sample of segmentation and linguistic representation of the segmented input.

height of first few input episodes. To compute it, we delay the recognition till a few episodes are written, find the maximum height of recorded items, and set SHU equal to 0.1 of this value. This way, we dynamically adapt the length descriptions to writer's script size and use 0.1 of her/his writing size as our description precision. To specify this value using linguistic terms, five fuzzy linguistic variables namely *ignorable*, *short*, *medium*, *long*, and *very long* are used.

Figure 9 presents a sample of one segmented input and its description using the above feature set. Note that Persian is written right to left.

### 3.3. Characters' pattern description

Character patterns are defined with the same descriptive language as the inputs, but to make the comparisons simpler, we have restricted pattern definitions' segment types just to lines, therefore, each character will be represented by a sequence of line segments, each having the previously stated three attributes (direction, curvature, and length). Note that type attribute is omitted as it is always line.

The database includes 60 samples, including the main body of all letters in all possible forms. It must be noted that as we ignore the dots and other additive marks and also not all letters have all four possible writing forms, 60 samples described the characters in their different common writing styles. Figure 10 presents some samples of character definitions.

## 4. Pattern Matching

Up to this point, we translated the input data and character patterns into fuzzy linguistic expressions. This section presents our method to compare these expressions.

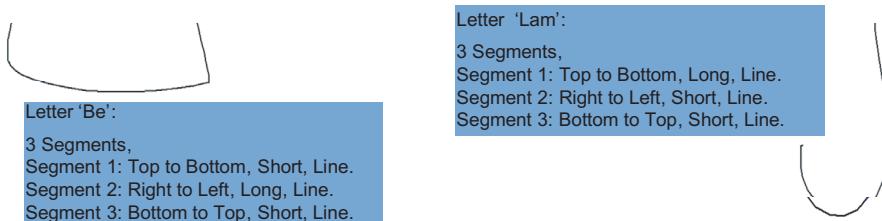


Fig. 10. Some samples for character pattern definitions, again note that Persian is written from right to left.

The major necessities of this part are:

- Coping with perturbations due to segmentation uncertainties.
- Ability to compare pattern and inputs with different number of segments, due to over segmentation or under segmentation (Fig. 11).
- There might be extra ignorable parts in the written text that are due to writer's style such as serifs, etc. or writing perturbations.

To have a process of creating multiple hypotheses and proving/disproving them as stated in Sec. 2, we divided the task into two levels: at the lower level, one part of one pattern is compared with one segment of input and some measures of difference are computed using a comparison look-up table; and at the higher level, a complete pattern is matched against the whole input.

#### 4.1. Low level comparison — one segment and one pattern part

The aim of this subsection is to compare one part of input with one segment part. As shown in Fig. 12, there are cases in which input segment may not be exactly similar to the pattern part, but do partly match. For example, the input curve (A) of Fig. 12 can be assumed similar with pattern (B) but it has some extra length before the similar section, or the input arc (C) of the same figure has some similarity with pattern (D), but it also has some extra, non-similar length after the similar section. To make such comparisons, we define the comparison measure as a triple

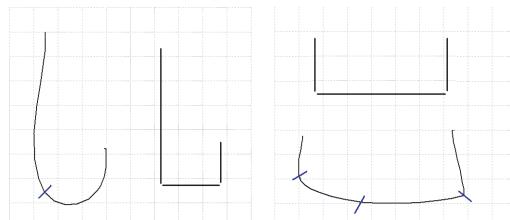


Fig. 11. A sample in which the input (the sketchy drawing) is separated into more/less number of parts from that of the pattern. The left input is called under-segmentation and the right one, over-segmentation.

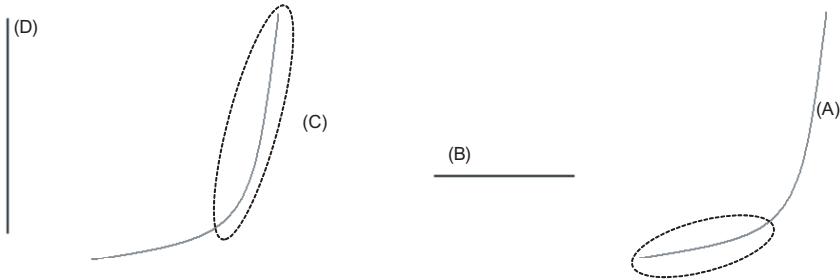


Fig. 12. Two samples for part to part comparison. (A) and (B) have some similar parts, preceded by some non-similarities in (A). (C) and (D) have a similar part, followed by a non-similar section in (C).

(*Extra Before, Matching Size, Extra After*) which specifies what percent of the two parts match, and how much extra did the input have before the matching part and how much did it have after the matching section. For example, input (A) of Fig. 12 matches with pattern Part (B) of the same figure with (70%, 30%, 0%) meaning that (A) has 30% similar Part with (B), but the similar section is preceded by 70% non-similar Part. Also, (C) matches (D) with (0%, 70%, 30%) meaning that (C) has 70% similar part with (D), followed by 30% non-similar part.

To make such comparisons, we have created a look up table, comparing a line segment with each possible input segment, along with their possible relative angles. Table 1 presents this table and Fig. 13 represents the visualization of these rules' if-parts. For example, third row of the table states that if the input part is an arc and it is in the same direction with the pattern part (which is a line), they match for 90% but the arc has 5% extra before and after the matching part. As another example, line 4 implies that a clockwise curved arc whose angle is one unit more than the line's angle (for example, the line is top to bottom while the arc is top-left to bottom-right) match in 75% of arc's length, but the arc has 25% extra after the matching part.

#### 4.2. High level comparison — an input and a pattern

The second layer of comparison is on top of the primary comparisons of pattern and input parts. To do so and to compare a complete input, i.e. a sequence of segments, with a pattern, or a sequence of pattern parts, we treat the pattern sequence as an elastic definition for the character that can stretch or skew based on the requirements of the input sequence. To do so, we use a non-deterministic finite state automaton with some modification where the sequence of input segments will be regarded as machine's inputs and pattern parts are machine's states (Fig. 14).

But the input is not necessarily segmented the same as the given abstract pattern. Figure 15 presents three other possible segmentations for different writings that all indicate the pattern of Fig. 14. As presented there, the required machine for these inputs differs from that of Fig. 14 and therefore, we need a structure that would be flexible enough to overcome all such possibilities.

Table 1. Single part comparison rules for line patterns. This table is visualized in Fig. 13.

Input Type	Angle Difference <sup>a</sup>	Extra Before	Matching	Extra After
1 Line	0	0%	100%	0%
2 Line	+1 or -1	25%	50%	25%
3 Arc	None	5%	90%	5%
4 Arc-C.	+1	0%	75%	25%
5 Arc-C.C.	+1	25%	75%	0%
6 Arc-C.	-1	0%	75%	25%
7 Arc-C.C.	-1	0%	75%	25%
8 Semicircle	None	15%	70%	15%
9 Semicircle-C.	+1	0%	50%	50%
10 Semicircle-C.	-1	50%	50%	0%
11 Semicircle-C.C.	-1	0%	50%	50%
12 Semicircle-C.C.	+1	50%	50%	0%

<sup>a</sup>Their difference based on our direction definitions. For example, top to bottom angle has one difference with either of {top left to bottom right} and {top right to bottom left}.

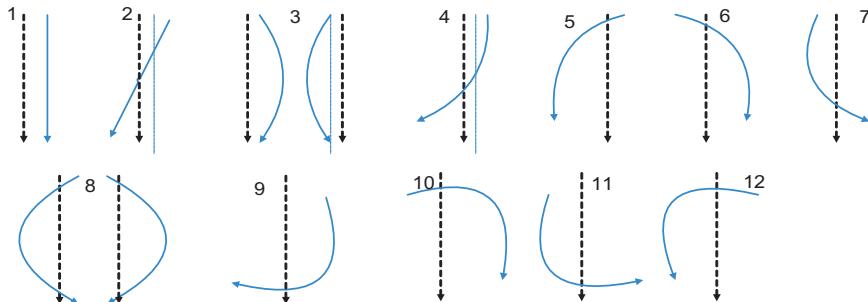


Fig. 13. Visualization of comparison rules for one line pattern (the dashed arrow) and different input segments.

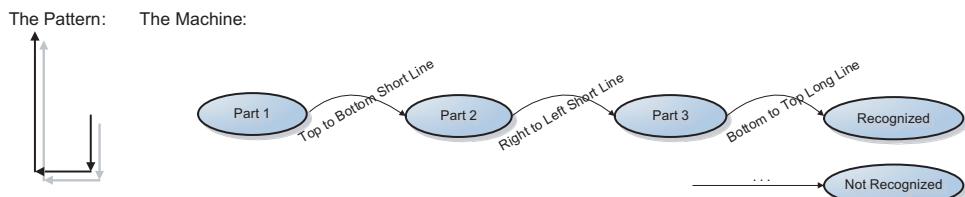


Fig. 14. Sample recognizer machine for an abstract pattern. Note that the other inputs that lead to failure in recognition are not drawn.

To overcome these problems, our machine has three major differences with conventional None Deterministic Finite State Automata (NDFSA)<sup>21</sup>:

- (i) We maintain a set of current possible states instead of one single current state; each current possible state has a degree of belief and some other properties.

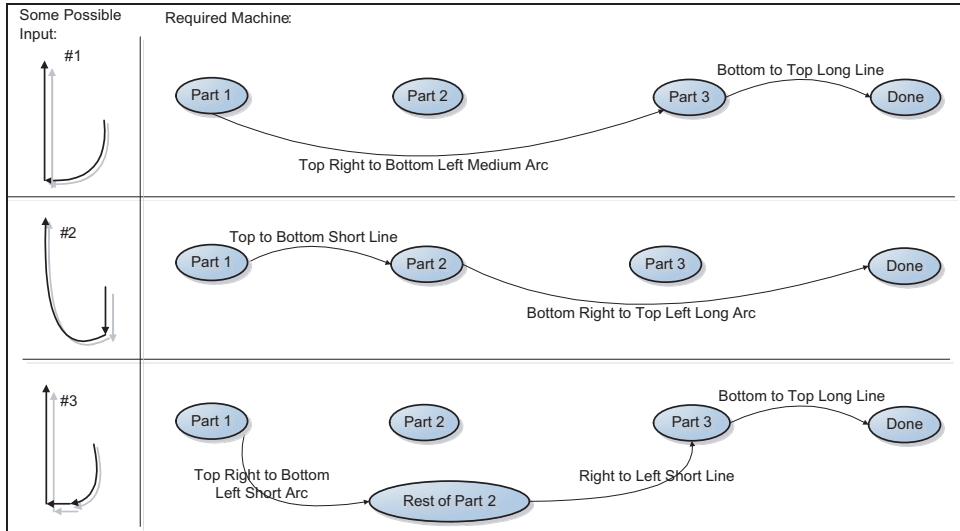


Fig. 15. Three sample inputs for the abstract pattern of Fig. 14.

- (ii) Instead of defining strict state transfer rules, we have a set of templates for state transfer where each pattern has some perquisites.
- (iii) Once a state loses certain capabilities, it is pruned out of a set of current possible states.

#### 4.2.1. Set of current possible states

In a conventional NDFSA, there are usually several choices for the next state given a certain input, and if the machine is set to process the inputs chronologically, it has to choose one of these possibilities randomly.<sup>21</sup> To overcome this unknown random factor, we maintain all possible hypotheses about the meaning of the input sequence given so far, proceed with all possible paths by keeping all of them as the set of Current States, and prune the set only when some states are falsified. The set of current possible states has just one member when the machine starts and once each input segment is given to the machine, all current possible states will produce their next possible state(s) and the set of all these next states will be the next step's possible states.

#### 4.2.2. Parameterized state definition

As the input is fed into the machine, different hypotheses are made on the meaning of different parts of it. Each of these hypotheses may have certain properties such as what length of the pattern is matched so far or how good is it matched here. To express these parameters, a state is defined as a five-dimensional vector (*POS*, *BEL*, *NOIS*, *CM*, *CF*) standing for *Current Position*, *Previous Belief*, *Previous*

*Noise, Current Match, & Current Flaw.* Table 2 presents the description of these fields.

#### 4.2.3. Templates for state transfer rules

To describe state transition rules for the pattern matching machine, let us first study another sample (Fig. 16 and Table 3). To match the input sample of Fig. 16 with the abstract given pattern, we can consciously follow this path of matching:

- (i) First, we need a top to bottom short line, but the first input segment is a right to left very small arc and they do not match at all. So, we skip the first input segment and assume that it is a non-matching length, adding 1 unit to noise.
- (ii) We still need to match Part 1 of the pattern, and input segment 2 well matches pattern Part 1. Thus, we accept it for pattern Part 1, and move the focus of reading to pattern Part 2.

Table 2. The descriptions of state properties.

Property	Description
Current Position	The machine is now accepting this part of the pattern, therefore, it ranges from 1 to last pattern part.
Previous Belief	The matching percentage of previous parts of the pattern. For example, we might be accepting Part 3 and have totally matched Parts 1 and 2 with 75% match.
Previous Noise	The total same of non-matching parts till current point. This measure is expressed in SHU.
Current Match	How much of the input has matched the current pattern part, in SHU
Current Flaw	How much of the input has not matched the current part, in SHU.

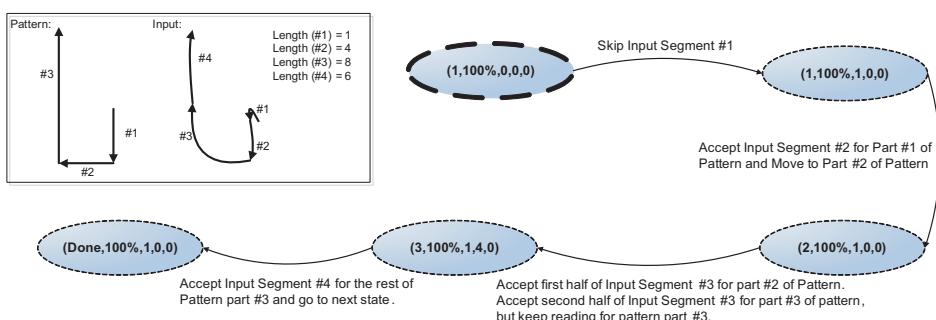


Fig. 16. A sample of input and abstract sample.

Table 3. Low level comparison table for Fig. 16. Each entree is (Extra Before, Matching Size, Extra After).

Input ↓	Pattern →	#1	#2	#3
#1		(0.5, 0, 0.5)	(0.2, 0.6, 0.2)	(0, 0.2, 0.8)
#2		(0, 4, 0)	(2, 0, 2)	(2, 0, 2)
#3		(4, 0, 4)	(0, 4, 4)	(4, 4, 0)
#4		(3, 0, 3)	(3, 0, 3)	(0, 6, 0)

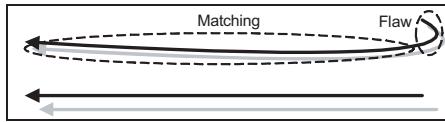


Fig. 17. Matching length and flaw.

- (iii) Then comes input segment 3, which partly matches pattern Part 2, but it has a tailing that does not match pattern Part 2. But the extra tail matches pattern Part 3 while it is still short for it. Therefore, we accept input segment 3 as satisfying pattern Part 2 and some of pattern Part 3 and wait for more for pattern Part 3.
- (iv) Finally, input segment 4 matches pattern Part 3, but it is short. Noting that we had an extra tail from input segment 3 which matched pattern Part 3 and that was short too, we can add these two parts and satisfy pattern Part 3 and state that the input and pattern are matched.

To express the above path following formally, we can first construct low level comparison table based on the triples of Sec. 4.1 which states how each segment of input matches each part of the pattern (Table 3). Then, following the state transitions of Fig. 16, we see that we have three classes of state transitions:

- SKIP: This is the action that happened at the first step of the above sample and happens when we want to ignore an input segment and add the flaws. Therefore, it changes the current state from  $(POS, BEL, NOIS, CM, CF)$  to  $(POS, BEL, NOIS, CM, CF + Length\ of\ (I))$ .
- ACCEPT, FORWARD MARCH: In this class of actions, the machine matches the given input with the current pattern part and moves to a next pattern part (what happened at step (ii) of previous example). Once this is done, the *Matching Size* of the current input is added to *Current Match* and its *Extra Before* and *Extra After* are added to *Current Flaw* as they do not take part in matching process and are not desired (see Fig. 17 for a sample).

But there are cases in which, the *Extra After* part matches the next pattern part and we must retain it, and this can even follow for a sequence of more than one pattern part. For example, as shown in Fig. 18, the input matches the first part

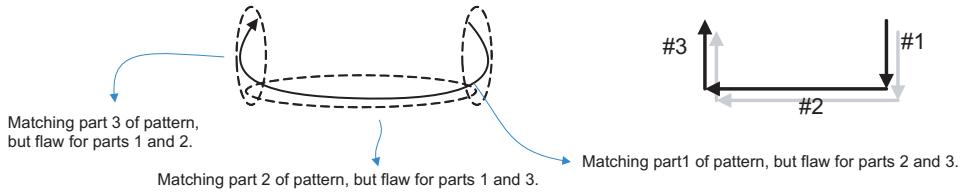


Fig. 18. Sample for forward marching more than one pattern part. The input segment matches partly each pattern part, and it can match the whole pattern.

of the pattern, but it has a long extra which matches the next pattern parts. In such cases, the machine can accept and pass all matching pattern parts and move to the next pattern part. The requirement of such transfers is that each of the matching patterns must have *Extra After* and *Extra Before* except the first and last parts where the first does not need *Extra Before* and the last does not need *Extra After*.

To express this formally, we use notation (3) and move from state  $POS$  to state  $POS + n$  when reading  $I_i$  with formula (4) limitations and effects. It must be noted that if several ( $n$ )s satisfy the *if* part, all of them are applied and the results are added as next possible states.

$$\left\{ \begin{array}{l} \text{Input Segments : } I_1, \dots, I_N \\ \text{Pattern Parts : } P_1, \dots, P_M \\ \text{CurrentState : } (POS, BEL, NOIS, CM, CF) \\ \text{Length}(I_x) : \text{The length of input segment } x \text{ in SHU units.} \\ \text{LenMatch}(x, p) : \text{The membership value of } x \text{ in fuzzy length specified} \\ \quad \text{for pattern Part } p. \\ \text{LowMatch}(I_i, P_j) = (EB_{i,j}, MS_{i,j}, EA_{i,j}), \text{based on low level comparisons,} \\ \quad \text{representing Extra Before, Matching Size and Extra After.} \end{array} \right. \quad (3)$$

$$\left\{ \begin{array}{l} \text{if} \\ \quad \left\{ \begin{array}{l} POS + n \leq M + 1 \\ n > 0 \\ [n = 1] \vee \\ [(\forall x; POS < x < POS + n | EB_{i,x} > 0) \wedge \\ (\forall x; POS \leq x < POS + n - 1 | EA_{i,x} > 0)] \end{array} \right. \\ \text{then add to Next Possible States:} \\ \quad \left\{ \begin{array}{l} (POS + n, \\ \quad BEL \times \text{LenMatch}(CM + MS_{i,cp}) \\ \quad \times \prod_{cp < x < cp+n} \text{LenMatch}(MS_{i,cp+x}) \\ \quad NOIS + EB_{i,CP} + EA_{i,CP+n-1}, \\ \quad 0, 0) \end{array} \right. \end{array} \right. \quad (4)$$

$$\begin{aligned}
 & \text{if} \\
 & \left\{ \begin{array}{l} n \geq 0 \\ POS + n \leq M \\ [n \leq 1] \vee \\ [(\forall x; POS < x \leq POS + n | EB_{i,x} > 0) \wedge \\ (\forall x; POS \leq x \leq POS + n - 1 | EA_{i,x} > 0)] \end{array} \right. \\
 & \text{then add to Next Possible States:} \\
 & \left\{ \begin{array}{l} (POS + n, \\ BEL \times LenMatch(CM + MS_{i,cp}) \\ \times \prod_{cp < x < cp+n} LenMatch(MS_{i,cp+x}) \\ NOIS + EB_{i,CP} + CF \times [n \neq 0] \\ + EA_{i,CP+n-1} \times [n = 0], \\ MS_{i,cp+x}, \\ CF \times [n = 0] + EA_{i,CP}) \end{array} \right. \tag{5}
 \end{aligned}$$

- ACCEPT, FORWARD MARCH, WAIT: As seen in the third step of Fig. 16 sample, there are cases where one input segment satisfies one or more pattern parts, and its tail partly satisfies the next pattern part. These cases are very much the same as the previous case (ACCEPT & FORWARD MARCH) but have a minor difference, formula (5) presents such transitions. Note that the bracket signs in the above formula are Iversonian brackets where  $[x]$  is 0 when  $x$  is false and is 1 when  $x$  is true.

#### 4.2.4. State pruning

Once a new state in  $(POS, BEL, NOIS, CM, CF)$  form is added to a set of next possible states, it is checked to see if its  $BEL$  is above a certain threshold and its  $NOIS$  is below another threshold and if the condition was not met, the state is thrown away and is not added to the next possible states. This is done to remove wrong hypotheses as soon as they are known to be not acceptable.

### 4.3. The pattern recognition approach all put together

Based on the definitions in Secs. 4.1 and 4.2, the complete pattern recognition system works as follows: once an episode of input is entered, segmented and described, the algorithm compares it with all stored character patterns and the pattern that gains the maximum similarity is chosen. To make the comparison for each pattern, *Compare Pattern* algorithm (Fig. 19), the current possible states are initialized with a single starting state  $(1, 100, 0, 0, 0)$ . Each input segment transfers all *Current Possible States* to *Next Possible States* and during this transfer, the states that fail some validation rules are pruned. If during this comparison, a state reaches the end of the pattern while the input sequence is not finished yet, it can be assumed that the input episode may cover more than one pattern. Thus, these states are separated and at the end, the rest of the input is checked to match any other pattern(s).

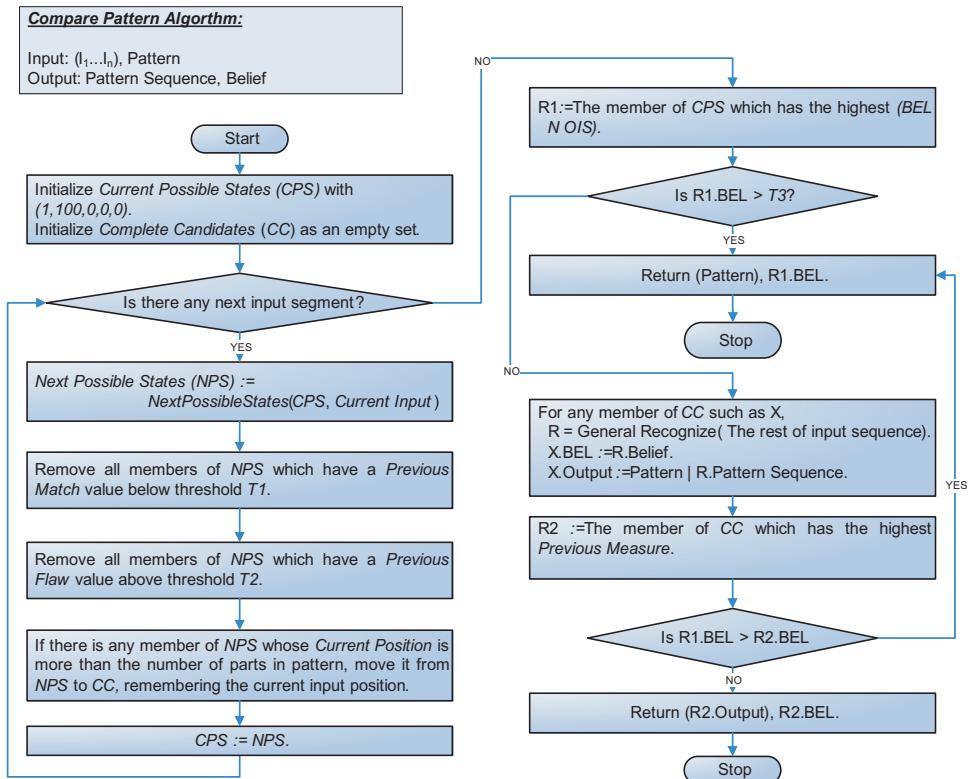


Fig. 19. Compare pattern algorithms.

Production of *Next Possible States* is also done through applying all possible state transfer templates on all *Current Possible States*. This is illustrated in Fig. 20.

## 5. Experimental Results

As there is no generally accepted benchmark to test the proposed algorithm for Persian handwriting and the test cases on none of the cited contributions for Persian or Arabic are available, we chose our test set from the first primary school book of Iranian schools which includes around 1250 words.<sup>b</sup> The test cases where written by 20 individuals where half of them were primary school kids and the other half were adults. The tests were also run once with a dictionary (the system knew the set of possible words) and once without dictionary. The test set was also simplified by removing the extra parts of the letters such as dots and other additives of Persian font. Table 4 presents the results.

<sup>b</sup>The list of words is available at <http://ce.sharif.edu/~halavati/Neveshtar/PrimarySchoolWords.txt>

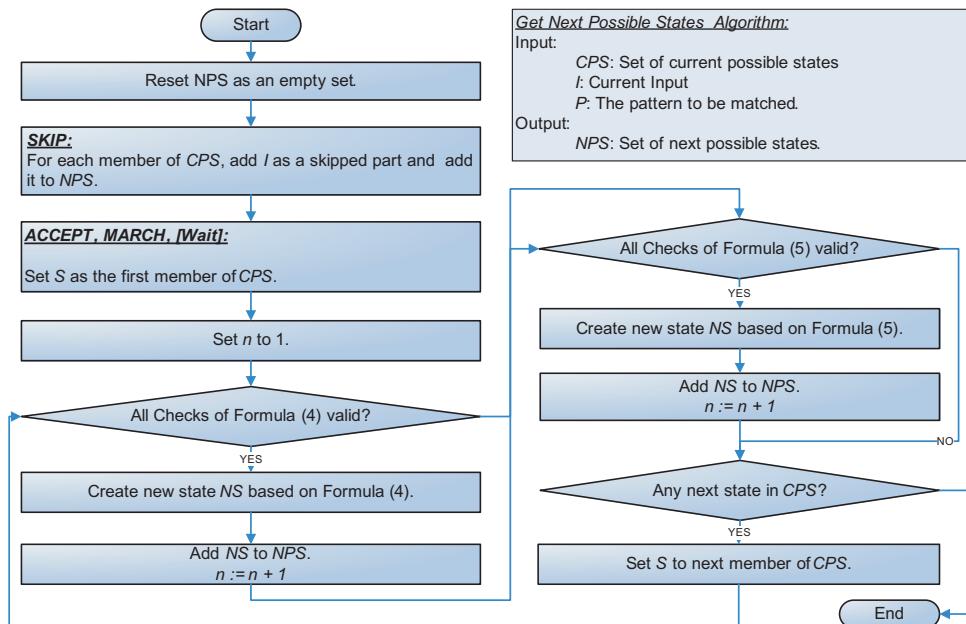


Fig. 20. Next possible state generator algorithm.

Table 4. Experimental results.

Writers Set	Without Dictionary	With Dictionary
Only Children	85%	98%
Only Adults	71%	92%
Both Together	78%	96%

As it was expected, the system has been notably more successful in recognizing writings of children as they are more faithful to writing rules. The system has successfully recognized children's writing in 98% of cases when the set of words was previously known and scored 85% when the input set was not previously mentioned. The results are 6% to 14% worse for grown up writers with and without dictionary due to their innovations in writing and having different styles.

Also to check the robustness of final recognition system to segmentation variations, we made a set of tests with a randomly chosen set of previous samples. To do so, the system was tested with 20 randomly generated parameter sets. In each set, the parameters were generated using a normal distribution whose averages were based on the values that we had used in previous experiments and the variances were 50% of average value (for example, the limit for selection of line in Part 3 was generated with  $\mu = 20 \sigma = 10$ ). The recognition results had only 2% variance (Table 5). Therefore, we can conclude that this approach does not need a perfect

Table 5. Segmentation variation test results.

Line Threshold	Arc Threshold	Results
$\mu = 20, \sigma = 10$	$\mu = 7, \sigma = 3.5$	$\mu = 82, \sigma = 2$

tuning of parameters but there exist a set of best parameters, at least for a set of specific writers.

And to check the memory requirements of recognition during the hypothesis generation phase, we computed the average and maximum number of concurrent hypotheses and the results were 10 and 34, respectively indicating that the pruning rules did not let the number of hypotheses increase exponentially, as it could happen theoretically.

## 6. Comparisons with Other Approaches

In this section, we compare our approach's features and results with that of previous contributions on recognition of Persian or Arabic handwritings.

### 6.1. Hierarchical rule-based approach

In Ref. 11, El-Sheikh and El-Taweel presented an approach based on a hierarchical tree of rules based on the number of strokes in each segment of the written text while the text is segmented into characters before being processed by their algorithm. They have reported 100% recognition accuracy for their test cases and no specific description of training/test data is provided. Putting aside the assumption that a neat input to character segmentation algorithm must exist before application of their method, the approach is highly sensitive to writing perturbations as the rule's hierarchy is based on the number of strokes in each character and this can be a variable based on writers' styles and friction of writing device.

### 6.2. Structural and fuzzy approach

In Ref. 8, a hybrid system of structural and fuzzy techniques is presented, that receives segmented handwritten characters, specifies them with some fuzzy descriptive language and then, a handmade fuzzy rule base chooses the correct class of letter. They have claimed 100% recognition rate while their test set is not mentioned. Similar to previous approach, they need a neat input to character segmentation approach beforehand and they are also highly sensitive to additional or missing parts of the pattern.

### 6.3. Template matching and dynamic programming

Alimi and Ghobel,<sup>6</sup> presented an approach for recognition of isolated Arabic handwritten characters in which, the input data is compared with the set of character

prototypes after smoothing, normalization and coding. They have reported 96% recognition accuracy with 1 writer. Again the input is assumed to be neatly segmented. Also, the dynamic pattern matching is dot to dot, creating a very big search space and also sensitivity to writers' styles.

#### **6.4. K-nearest neighbor classification**

This approach<sup>10</sup> uses a K-Nearest Neighbor classifier system on a set of stable features such as the number of dots, relative position of dots, and number and position of other secondary strokes. The approach has resulted in 84% accurate recognition with seven writers. Initial character segmentation is preassumed and the approach is sensitive to addition/removal of any part.

#### **6.5. Evolutionary neuro-fuzzy approach**

Alimi<sup>7</sup> proposed a complete system that segments and recognizes input sequence using a set of six feature vectors. The input is segmented using a genetic process into letters and letters are identified using a fuzzy radial basis function. The system scored 89% for a single writer. Although it performs a robust segmentation using genetic algorithm, it becomes quite slow in average and with long words, and it is writer dependant.

#### **6.6. Neural network approach**

Mezghani<sup>18</sup> and Klassen<sup>14,15</sup> proposed approaches based on multilayer perceptron and self-organizing map neural networks, both to classify the main body of isolated letters. The first approach resulted in 95% correct recognition for multiple writer cases and the second resulted in 86.56% for single writer tests.

#### **6.7. Comparison with our approach**

As summarized in Table 6, in comparison with other approaches, the first major advantage of our approach is its ability to segment and recognize at the same time, while only one other approach does this and it requires considerable amount of computations (evolutionary approach). Also, the approach is one of the rare methods that are not writer dependant and can tolerate writing perturbations. As the data sets of other methods are not present, we cannot make a direct comparison in recognition accuracy but as it can be generally grasped, we have used a much wider test set including many writers.

Putting all together, we can conclude that this approach can recognize the written text user independently, insensitive to perturbation or minor styles, without supervision (for segmentation), and with acceptable result without dictionary and very good result with dictionary (1250 words).

Table 6. Comparison with other methods.

Approach	Requires Segmentation	Sensitivity to Perturbations	Number of Tested Writers	Uses Dictionary	Results
Hierarchical Rule Base	Yes	High	?	?	100%
Structural Fuzzy	Yes	High	?	?	100%
Template Matching	Yes	Average	1	?	96%
K-Nearest Neighbor	Yes	High	7	?	84%
Evolutionary Neuro Fuzzy	No	Average	1	?	89%
Our Approach without Dictionary	No	Low	20	No	78%
Our Approach with Dictionary	No	Low	20	Yes	96%

## 7. Conclusions and Future Works

Recognition of Persian handwriting is a complicated task as Persian is fully cursive, characters are written in up to four different forms in different parts of the world, and people write in several different styles. The presented approach works on online data, segments the input set into a sequence of lines, arcs and half-circles and represents these segments with fuzzy linguistic terms. While the letter patterns are also defined with similar language, a flexible comparison algorithm compares the input sequence with different character sequences.

As there is no general benchmark on Persian handwriting, the approach is tested with the set of words that are used in first primary Iranian school books and the results have been quite satisfactory for grown up writers and much better for children as they obey writing rules more than grown ups. The algorithm has also been tested with different parameter sets and it has shown very little sensibility to its settings.

The major advantages of this algorithm are: it is fast, yet quite robust to writing perturbations, it does not need adaptation phase for different writers, and it can segment input into letters automatically.

As the next step to this contribution, we are adding circles as another input token, making use of dots and other additives of Persian writing in the recognition system, and designing an active learning method to improve the ability of automatically refining definitions and adopting new styles.

## Acknowledgments

We need to thank Ms. Mahdieh Soleymani and Mr. Reza Hesami Fard for their valuable discussions on different parts of this contribution and Dr. Mansour Jamzad for his support on project initiation.

## References

1. K. Abbasian and E. Karbir, Online recognition of Persian script, *Proc. 6th Nat. Conf. Electrical Engineering*, Tehran, Iran (1999), pp. 146–141.

2. I. S. I. Abuhaiba, M. J. J. Holt and S. Datta, Recognition of off-line cursive handwriting, *Comput. Vis. Imag. Process.* **71**(1) (1998) 19–38.
3. M. Aksela, Handwritten character recognition: a palm-top implementation and adaptive committee experiments, Master's thesis, Department of Engineering, Physics, and Mathematics, Helsinki University of Technology (2000).
4. S. Al-Emami and M. Usher, On-line recognition of handwritten Arabic characters, *IEEE Trans. Patt. Anal. Mach. Intell.* **12**(7) (1990) 704–710.
5. A. M. Alimi, A neuro-fuzzy approach to recognize Arabic handwritten characters, *Proc. IEEE Int. Conf. Neural Network* **3** (1997) 1397–1400.
6. A. Alimi and O. Ghorbel, The analysis of error in an on-line recognition system of Arabic handwritten characters, *Proc. ICDAR 1995* (1995), pp. 890–893.
7. A. Alimi, An evolutionary neuro-fuzzy approach to recognize on-line Arabic handwriting, *Proc. 4th Int. Conf. Document Analysis and Recognition (ICDAR '97)* (1997), pp. 382–386.
8. F. Bouslama and A. Amin, Pen-based recognition system of Arabic character utilizing structural and fuzzy techniques, *Proc. Second Int. Conf. Knowledge-Based Intelligent Electronic Systems*, eds. L. C. Jain and R. K. Jain (1998), pp. 76–85.
9. W. Doster and R. Oed, Word processing with online script recognition, *IEEE MICRO* **4** (1984) 36–43.
10. M. El-Wakil and A. Shoukry, On-line recognition of handwritten isolated Arabic characters, *Patt. Recogn.* **22**(2) (1989) 97–105.
11. T. S. El-Sheikh and S. G. El-Taweelel, Real time Arabic handwriting character recognition, *Patt. Recogn.* **24**(12) (1990) 1323–1332.
12. R. Halavati, S. H. Zadeh and Y. Mokri, How children read Persian?, *Sharif Uni. Technology Internal Report No 138412* (2004).
13. N. Joshi, G. Sita, A. G. Ramakrishnan and S. Madhvanath, Comparison of elastic matching algorithms for online Tamil handwritten character recognition, *Proc. Ninth Int. Workshop on Frontiers in Handwriting Recognition (IWFHR'04)* (2004).
14. T. J. Klassen, Toward neural network recognition of handwritten Arabic letters, Master of Computer Science Thesis, Dalhousie University, 2001.
15. T. J. Klassen and M. I. Heywood, Toward the on-line recognition of arabic characters, *Proc. 2002 Int. Joint Conf. Natural Networks (IJCNN'02)*, **2** (May 2002), pp. 1900–1905.
16. A. Malaviya and R. Klette, A fuzzy syntactic method for on-line handwriting recognition, *Advances in Structural and Syntactical Pattern recognition, SSPR'96*, Lecture Notes in Computer Science, Vol. 1121 (1996), pp. 381–392.
17. E. Mandler, Advance preprocessing technique for online script recognition of non connected symbols, *Proc. 3rd Int. Writing Computer Applications* (1987), pp. 64–66.
18. N. Mezghani and M. Cheriet, Combination of pruned Kohonen maps for online Arabic characters recognition, *Proc. 7th Int. Conf. Document Analysis and Recognition (ICDAR'03)* (2003), pp. 900–904.
19. R. Ranawana, V. Palade and G. E. M. D. C. Bandara, An efficient fuzzy method for handwritten character recognition, *Proc. KES2004* (2004), pp. 698–707.
20. S. M. Razavi and E. Kabir, Online recognition of Persian isolated characters, *Proc. 3rd Nat. Conf. Intelligent Systems, Kerman, Iran* (2005).
21. M. Sipser, Introduction to the theory of computation, PWS, Boston (1997), pp. 47–63.
22. C. C. Tappert, Cursive script recognition by elastic matching, *IBM J. Res. Develop.* **26**(6) (1982) 765–771.

23. C. C. Tappert and J. M. Kutzberg, Elastic matching for handwritten symbol recognition, *Proc. IBM Int. Conf. Image Processing and Pattern Recognition* (1978).
  24. B. Wan, An interactive mathematical handwriting recognizer for the pocket PC, M.S. thesis, Department of Computer Science, University of Western Ontario (2002).
- 



**Ramin Halavati** received his M.S. in artificial intelligence from Sharif University of Technology, (Tehran, Iran) in 2003, and is now pursuing the Ph.D. in artificial intelligence at the Computer Engineering Department of Sharif University of Technology.

His research interests include cognition, learning and consciousness, and his Ph.D. dissertation is on learning in pattern recognition.



**Saeed Bagheri Shouraki** received his Ph.D. in fuzzy control systems from the University of Electro-Communications (Tokyo, Japan) in 2000. He is now at the Sharif University of Technology (Tehran, Iran).

He has published more than 90 articles covering his research interests that include control, robotics, artificial life and soft computing.