

باسمه تعالی

پروژه درس پردازش تصاویر دیجیتال

تشخیص سلسله مراتبی میکروآنوریسم‌ها در تصاویر شبکه با بکارگیری فیلترینگ

همبستگی چندمقیاسه^۱

تهیه و تنظیم:

بهاره بافنده مایوان-۹۳۱۶۴۸۱۱۲۷

چکیده:

شبکیه‌ی دیابتی از عوارض جدی و رایج ناشی از دیابت بلند مدت در افراد می‌باشد. این عارضه باعث آسیب رساندن به شبکه و در نهایت منجر به کوری فرد می‌شود. از آنجایی که ظهور ضایعات قرمز رنگ (میکروآنوریسم‌ها) بر روی شبکه از اولین نشانه‌های شبکیه‌ی دیابتی محسوب می‌گردد، تحقیقات گسترده‌ای بر روی تشخیص موثر و مکان‌یابی صحیح این نابهنجاری‌ها صورت گرفته است. در مقاله‌ی [۱] که قصد تحلیل و پیاده‌سازی آنرا داریم برخلاف الگوریتم‌های موجود از روشی جدید به نام فیلترینگ همبستگی چندمقیاسه و آستانه‌گذاری پویا، بهره گرفته شده است. این روش شامل دو مرحله می‌شود: ۱- تشخیص ضایعات قرمز رنگ کاندید و ۲- تشخیص دقیق ضایعات قرمز رنگ. دیدگاه ارائه شده در این مقاله با استفاده از داده‌های موجود در وب سایت ROC competition مورد ارزیابی قرار گرفته است و کارایی و موثر بودن روش ارائه شده اثبات گردیده است.

¹ Multi-Scale Correlation Filtering

فهرست مطالب

فصل ۱: مروری بر مقاله و روش انجام کار	۳
مقدمه	۳
معمول‌ترین روش‌های غربال‌گری شبکیه‌ی دیابتی	۵
تشخیص خودکار شبکیه‌ی دیابتی	۶
مرحله‌ی اول (تشخیص ضایعات قرمز رنگ کاندید)	۸
مرحله‌ی دوم (تشخیص ضایعات قرمز رنگ درست)	۱۴
نتایج بدست‌آمده از آزمایش‌ها و تحلیل آن‌ها	۱۷
جمع‌بندی	۲۰
فصل ۲: مروری بر پیاده‌سازی مقاله و نتایج بدست‌آمده	۲۲
مقدمه	۲۲
نتایج	۲۳
پیوست	۲۷
مراجع	۴۷

فصل ۱: مروری بر مقاله و روش انجام کار

مقدمه:

تصاویر عمقی^۲ از چشم که تحت عنوان تصاویر شبکیه نیز شناخته شده‌اند می‌توانند اطلاعات مفیدی راجع به شبکیه، چشم‌درد و یا حتی بیماری‌های بدن انسان از قبیل دیابت، فشارخون، آب‌سیاه، تصلب شرائین و انسداد شرائین شبکیه در اختیار ما قرار بدهند. یکی از مهمترین کاربردهای تصاویر شبکیه، تشخیص شبکیه‌ی دیابتی است که از نتایج دیابت بلندمدت بوده و منجر به شکل‌گیری ضایعاتی بر روی شبکیه می‌گردد که نهایتاً منجر به کوری فرد می‌شود.

² Ocular Fundus

برای جلوگیری از تاثیر این عارضه‌ی جدی بر روی بینایی فرد بیمار، تشخیص به موقع شبکه‌ی دیابتی و درمان مناسب آن از اهمیت بالایی برخوردار است. همچنین برای ارزیابی میزان موثر بودن درمان‌های پزشکی صورت گرفته و مشاهده‌ی میزان پیشرفت بیماری، نظارت بر روی روند بیماری امری مهم و حیاتی می‌باشد. و این مهم تنها از طریق تشخیص و طبقه‌بندی تغییرات در تصاویر شبکه که در مقاطع زمانی مختلفی از چشم گرفته شده‌اند میسر می‌باشد.

عموما شبکه‌ی دیابتی می‌تواند به ۴ مرحله دسته‌بندی شود: شبکه‌ی دیابتی غیرتوسعه یافته‌ی خفیف^۳، شبکه‌ی دیابتی غیرتوسعه یافته‌ی میانه^۴، شبکه‌ی دیابتی غیرتوسعه یافته‌ی جدی^۵ و شبکه‌ی دیابتی توسعه یافته^۶. اگر شبکه‌ی دیابتی در مراحل اولیه تشخیص داده شده و درمان شود، آسیب آن به حداقل ممکن رسیده و از کم شدن بینایی به صورت گسترده‌ای جلوگیری به عمل می‌آید. بنابراین آزمایش شبکه‌ی فرد بیمار از اهمیت بالایی برخوردار است. اما اگر تشخیص شبکه‌ی دیابتی به صورت دستی توسط پزشک صورت بگیرد عملی زمان‌بر بوده و با خطاهای انسانی مواجه است. از این رو تحلیل‌های کامپیوتری و اتومات شبکه‌ی فرد مبتلا به دیابت به چشم‌پزشک کمک می‌کند تا جمعیت بزرگتری از بیماران را با دقت بالاتری مورد ارزیابی قرار دهد. به کمک پردازش تصاویر می‌توان در زمینه‌ی بهبود تصاویر و استخراج ویژگی‌ها^۷، غربال‌گری جمعی^۸ و کشف تغییرات (عیب‌شناسی) و همچنین طبقه‌بندی (نظارت)^۹ ابزارهایی قدرتمند تهیه کرد.

³ Mild Non-Proliferative Diabetic Retinopathy (NPDR)

⁴ Moderate NPDR

⁵ Severe NPDR

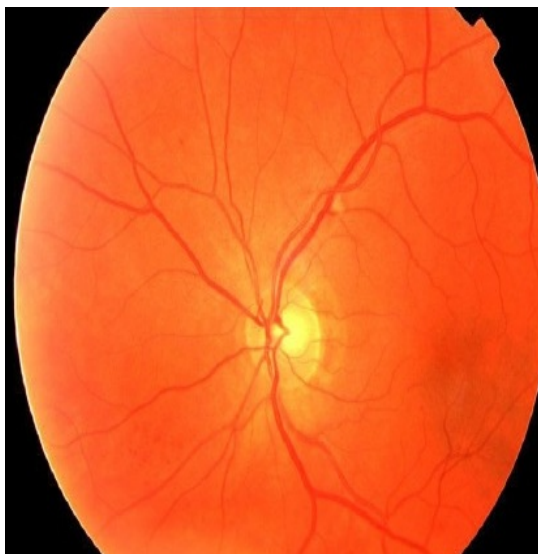
⁶ Proliferative Diabetic Retinopathy (PDR)

⁷ Image enhancement and feature extraction (feature based image registration)

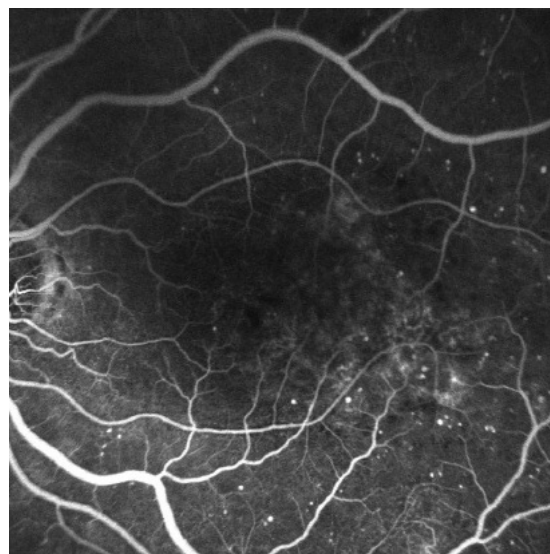
⁸ Mass screening (diagnosis)

⁹ Change detection, and classification (monitoring)

معمول‌ترین روش‌های غربال‌گری شبکیه‌ی دیابتی:



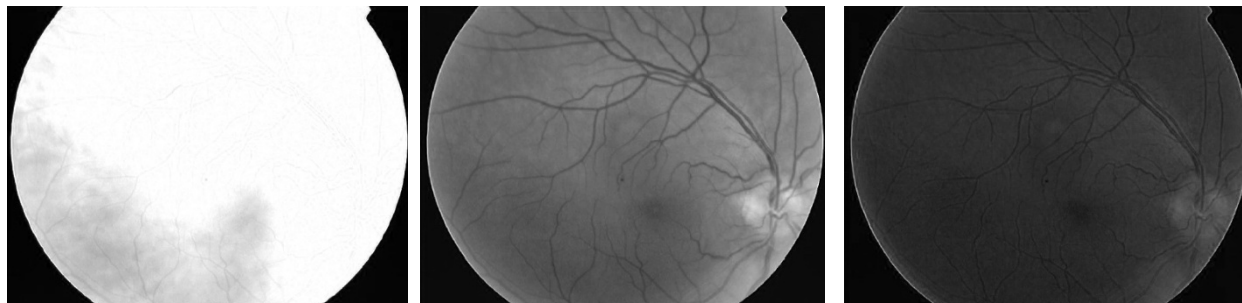
شکل ۱- ب) تصویر رنگی شبکیه



شکل ۱- الف) روش آنژیوگرافی فلورسین

آنژیوگرافی فلورسین، ابزاری تخمینی و پزشکی برای غربال‌گری شبکیه‌ی دیابتی می‌باشد. در این روش قبل از تصویربرداری از شبکیه، ماده‌ای به نام فلورسین به بدن تزریق می‌شود. این ماده باعث می‌شود تا رگ‌ها خودشان را به صورت واضح‌تری در عکس نشان بدهند. همانطور که در شکل ۱-الف می‌توان دید تصاویری که از این طریق از سطح شبکیه گرفته می‌شوند سطح خاکستری بسیار واضحی دارند و در تشخیص خونریزی‌ها و رشد رگ‌های جدید به ما کمک زیادی می‌کنند. اما این روش بدیل تزریق ماده‌ی فلورسین به بدن، مورد علاقه‌ی بیماران نمی‌باشد. در مقابل تحلیل تصاویر رنگی شبکیه یک روش معقولانه است چراکه تهیه آن ساده و سریع بوده و ماهیتی غیر تزریقی دارد. در پردازش تصاویر رنگی شبکیه عموماً صفحه‌ی سبز آن را مورد توجه قرار می‌دهیم چراکه ضایعات قرمز رنگ در آن دارای بیشترین کنتراست با زمینه می‌باشند. در شکل ۲ صفحات قرمز (I-red)، سبز (I-green) و آبی (I-blue) از تصویر رنگی شبکیه نشان داده شده‌است. همانطور که مشاهده

می‌شود در تصویر صفحه‌ی سبز شبکیه، عناصری مانند رگ‌های خونی، ضایعات، صفحه‌ی نوری و غیره بیشترین وضوح را دارند.



شکل ۲-ج) صفحه قرمز

شکل ۲-ب) صفحه سبز

شکل ۲-الف) صفحه آبی

تشخیص خودکار شبکیه‌ی دیابتی

در مقاله‌ی [۲] برای تشخیص خودکار شبکیه‌ی دیابتی از شبکه‌های عصبی مصنوعی استفاده شده‌است. شبکه‌ی عصبی مورد نظر در این روش شامل لایه‌ای از پیکسل‌های ورودی، لایه‌ی مخفی و لایه‌ی خروجی می‌باشد و برای تشخیص مشخصه‌هایی نظیر رگ‌های خونی، ترشحات و خونریزی‌ها آموزش دیده است. بسته به مشخصه‌ی کشف شده، تصاویر به مربع‌های 20×20 و 30×30 تقسیم می‌شوند و سپس یک مشاهده‌گر آموزش دیده‌شده مربع‌ها را به چهار دسته طبقه‌بندی می‌کند. دسته‌ی اول قسمتی از شبکیه‌ی عادی که رگ‌های خونی در آن مشاهده نمی‌شوند، دسته‌ی دوم قسمتی از شبکیه‌ی عادی که رگ‌های خونی در آن مشاهده می‌شوند، دسته‌ی سوم قسمتی از شبکیه که ترشحات در آن مشاهده می‌شود و دسته‌ی چهارم قسمتی از شبکیه که خونریزی و ضایعات در آن مشاهده می‌شود. در این روش، دسته‌بندی خونریزی‌ها سخت‌ترین قسمت کار است چراکه مقادیر پیکسل‌هایشان مشابه رگ‌های خونی است. نرخ true positive در این روش ۸۸.۴ و نرخ true negative در آن ۸۳.۵ است.

در مقاله‌ی [۳] سیستم طراحی شده، برای تشخیص شبکه‌ی دیابتی، رگ‌های خونی، صفحه‌ی نوری، fovea و ضایعات تیره و روشن را از هم تمیز می‌دهد. ضایعات تیره رنگ شامل میکروآنوریزم‌ها و خونریزی‌ها و ضایعات روشن شامل ترشحات سخت که تحت عنوان نقاط cotton wool شناخته می‌شوند، می‌گردد. عملیات تشخیص بر پایه‌ی تعداد، نوع و موقعیت مکانی نواحی غیرمعمول نسبت به fovea صورت می‌گیرد. تصویر اصلی به دو قسمت تقسیم می‌شود: fovea و ناحیه‌ی باقی‌مانده از شبکه. در هر دو قسمت بدست آمده، نواحی غیرمعمول و یا به اصطلاح نابهنجاری‌ها با حذف ویژگی‌های شبکه‌ی عادی پدیدار می‌شوند. در واقع با حذف ویژگی‌های عادی از شبکه، چیزی که باقی می‌ماند همان نابهنجاری‌های موجود به اضافه‌ی زمینه‌ی شبکه است. سپس ویژگی‌های استخراج شده از اشیای تشخیص داده شده به یک شبکه‌ی عصبی خورانده می‌شوند تا آن‌ها را به دو دسته‌ی عادی و غیرعادی دسته‌بندی کند.

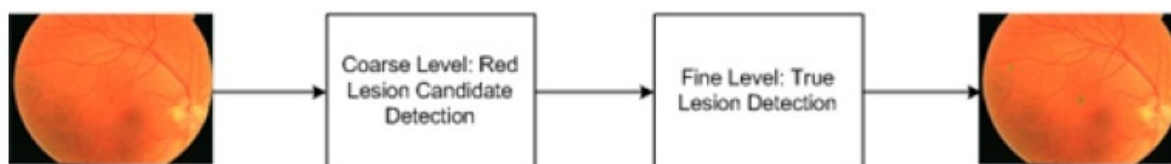
دو روش ذکر شده در بالا در برخی از موارد موثر هستند اما دارای محدودیت‌هایی هم می‌باشند از جمله اینکه با این روش‌ها میکروآنوریزم‌ها استخراج نمی‌شوند بلکه فقط دسته بندی می‌گردند و از طرف دیگر نیز استفاده از شبکه‌ی عصبی عملی زمانبر است.

در مقاله‌ی [۴] محققین بر این مشکلات فائق آمده و راهی برای شناسایی و استخراج ضایعات قرمز رنگ در تصاویر شبکه ابداع کردند که بر اساس کشف و دسته‌بندی عناصر کاندید عمل می‌کند. کشف عناصر کاندید شامل مکان‌یابی همه‌ی ضایعات قرمز رنگ ممکن می‌شود که به وسیله‌ی ترکیب روش ریخت‌شناسی ریاضی^{۱۰} [۵] با دسته‌بندی پیکسل‌ها انجام می‌گیرد. پیکسل‌ها با استفاده از یک طبقه‌بند k-NN با یک مرجع استاندارد که نیاز به نشانه‌گذاری دستی هر تصویر دارد طبقه‌بندی می‌شوند. سپس از کاندیدهای باقی‌مانده ۶۸ ویژگی

¹⁰ Math Morph

استخراج می‌گردد و دوباره با به‌کارگیری k -NN طبقه‌بندی می‌شوند. این روش هنگام تصمیم‌گیری در مورد اینکه آیا یک تصویر مشخص دارای ضایعه‌ی قرمز رنگ می‌باشد و یا نه، دارای نرخ true positive ۱۰۰ و نرخ false positive ۸۷ است. این روش با وجود اینکه نرخ true positive و false positive بالایی دارد نمی‌تواند تمام ضایعات ممکن در تصویر را تشخیص دهد و از طرفی طبقه‌بندی دستی پیکسل‌ها در آن عمل دشواری بوده و نیازمند متخصصین خبره در این زمینه است.

برای جلوگیری از این مشکلات در این مقاله روشی جدید و کاملاً خودکار ارائه شده‌است که در آن برای تشخیص ضایعات قرمز رنگ بر روی تصاویر شبکه‌ی از روشی سلسله‌مراتبی استفاده شده‌است. در این روش از فیلترینگ همبستگی چندمقیاسه و آستانه‌گذاری پویا، بهره گرفته شده‌است و شامل دو مرحله می‌باشد: ۱- تشخیص ضایعات قرمز رنگ کاندید و ۲- تشخیص دقیق ضایعات قرمز رنگ.

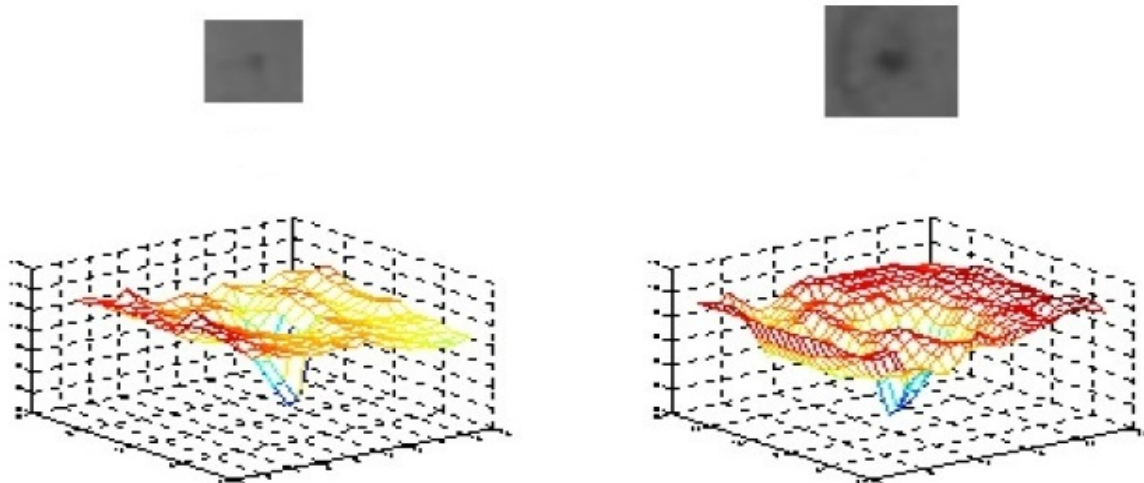


شکل ۳) الگوریتم ارائه شده در مقاله در یک نگاه

مرحله ۱: تشخیص ضایعات قرمز رنگ کاندید:

در این مرحله تمامی کاندیدهای مشکوک به ضایعه‌ی قرمز رنگ بر روی یک تصویر شبکه‌ی مشخص می‌شوند. در شکل ۴ دو ضایعه‌ی قرمز رنگ پیداشده در شکل ۱-ب را به همراه توزیع‌های سطح خاکستری متناظرشان نشان داده‌ایم. همانطور که در شکل مشخص است ضایعات قرمز رنگ شکلی گوسی دارند. با توجه به این نکته

می‌توان برای تشخیص ضایعات، از یک تابع گاوسی بر اساس میزان شباهت توزیع‌های سطح خاکستری‌شان استفاده کرد.

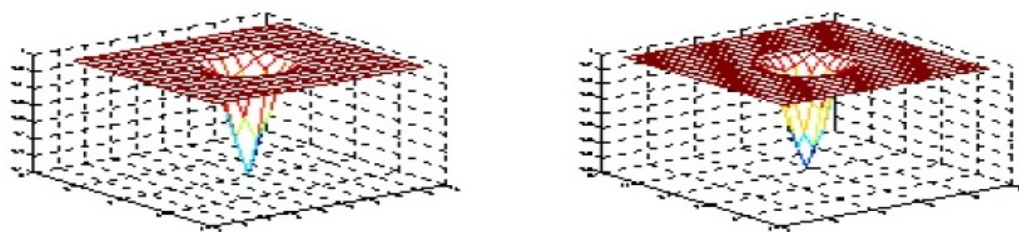


شکل ۴-الف) ضایعه قرمز رنگ ۱ به همراه توزیع سطح خاکستری آن شکل ۴-ب) ضایعه قرمز رنگ ۲ به همراه توزیع سطح خاکستری آن

تابع گاوسی می‌تواند به صورت زیر تعریف شود:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{\frac{-x^2+y^2}{2\sigma^2}}$$

توزیع این تابع را در شکل زیر نشان داده‌ایم:



شکل ۵) نمایشی از هسته‌ی گاوسی با فاکتورهای اسکال متفاوت ۱.۱ (سمت چپ) و ۱.۵ (سمت راست)

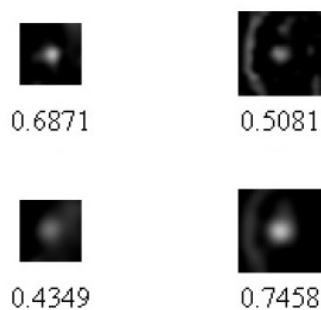
ضریب همبستگی راه خوبی برای اندازه‌گیری شباهت بین تابع گوسی و توزیع سطح خاکستری از ضایعات است. اگر این دو با یکدیگر شباهت داشته باشند، مقدار ضریب همبستگی بالا می‌رود و در صورت عدم شباهت این مقدار پایین خواهد آمد. ضریب همبستگی مقداری بین صفر تا یک دارد.

تابع همبستگی را به صورت زیر تعریف می‌کنیم:

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right) \left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}}$$

که در آن \bar{A} و \bar{B} میانگین هستند.

از آنجایی که اندازه‌ی ضایعات متغیر است، برای هسته‌ی گوسی مقادیر مختلفی از سیگما را نیاز داریم. برای رسیدن به این هدف و برای تطبیق پیدا کردن با ابعاد مختلف ضایعات، چندین مقیاس را انتخاب کرده‌ایم. شکل ۶ این موضوع را به خوبی نشان می‌دهد. در این شکل، تصاویر سمت راست مربوط به ضایعه‌ی قرمز رنگ ۱ (مشخص شده در شکل ۴) و تصاویر سمت چپ مربوط به ضایعه‌ی قرمز رنگ ۲ می‌شوند. دو تصویر بالا، واکنش‌ها و ضرایب همبستگی ماکزیمم مربوط به هسته‌ی گوسی با مقادیر کوچک و دو تصویر پایین، واکنش‌ها و ضرایب همبستگی ماکزیمم مربوط به هسته‌ی گوسی با مقادیر بزرگ را نشان می‌دهد.



شکل ۶) پاسخ ضایعات به هسته‌های گوسی مختلف

با توجه به شکل ۶ می‌توان به‌وضوح دید که بکاربردن تنها یک مقیاس، نمی‌تواند کشف ضریب ماکزیمم را تضمین کند. به طور مثال در مورد ضایعه‌ی قرمز رنگ ۱ با بکاربردن هسته‌ی گوسی با مقادیر بزرگ، ضریب بزرگتری ساخته شده‌است در حالیکه در مورد ضایعه‌ی قرمز رنگ ۲ با بکاربردن هسته‌ی گوسی با مقادیر کوچک، به ضریب بزرگتری رسیده‌ایم.

بنابراین اولین قدم از مرحله‌ی اول برای کشف کاندیدها، بکاربردن یک فیلتر غیرخطی با هسته‌های گوسی چندمقیاسه است تا به وسیله‌ی آن برای هر پیکسل از تصویر شبکه‌ی، یک ضریب همبستگی را حساب کنیم. به‌دلیل اینکه ضایعات قرمز رنگ، دایره‌ای شکل هستند برای تضمین اینکه واکنش حاصل (ضریب همبستگی) به مقدار بالایی خواهد رسید، هسته‌های گوسی را نیز دایره‌ای شکل انتخاب می‌کنیم. بر اساس آزمایشات گسترده‌ای که انجام شده‌است برای نمایش میکروآنوریسم‌هایی با اندازه‌های مختلف، پنج مقیاس را برای هسته‌ی گوسی انتخاب کرده‌ایم. سیگماهای انتخاب شده برای تابع گوسی به ترتیب برابر با ۱.۱، ۱.۲، ۱.۳، ۱.۴، ۱.۵ و ۱ در نظر گرفته شده‌اند. برای تشکیل پاسخ نهایی، ضرایب ماکزیمم از هر پنج پاسخ تولید شده باهم ترکیب می‌شوند. شکل ۷-ب نمونه‌ای از یک پاسخ نهایی را نشان می‌دهد. نقاط روشن‌تری که در شکل نشان داده شده‌اند ضریب بالاتری دارند و بنابراین احتمال اینکه ضایعات قرمز رنگ درستی باشند در آنها بیشتر است.



شکل ۷) پاسخ نهایی از یک تصویر آزمایشی (تصویر ورودی در سمت چپ و تصویر خروجی در سمت راست)

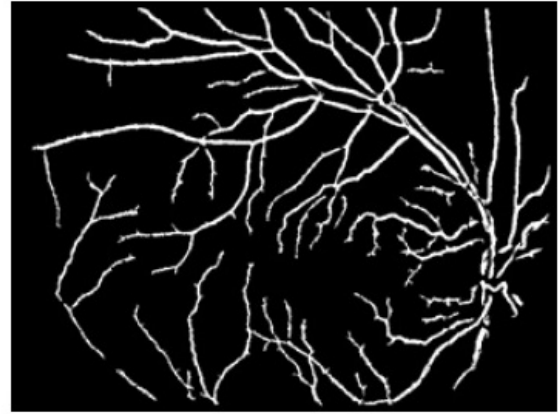
در مرحله بعد برای تعیین تعداد کاندیدهای تعیین شده به عنوان ضایعات قرمز رنگ، یک مقدار آستانه را به حاصل اعمال می‌کنیم. به طور مثال همانطور که در شکل ۸ نشان داده شده است برای قطعه‌بندی شکل ۷-ب آستانه‌ای با مقدار ۰.۴ در نظر گرفته شده و به تصویر اعمال شده است که خروجی باینری‌ای را برای ما تولید کرده است.



شکل ۸) نتیجه‌ی حاصل پس از قطعه‌بندی (مقدار آستانه را برابر با ۰.۴ در نظر گرفته‌ایم و نقاط سفید رنگ مشخص‌کننده‌ی

همه‌ی ضایعات قرمز رنگ ممکن هستند.)

از آنجایی که ضایعات قرمز رنگ نمی‌توانند روی عروق خونی رخ بدهند یک روش آستانه‌گذاری افقی برای مکان‌یابی عروق و حذف نقاطی که روی آن‌ها قرار دارند صورت می‌گیرد. بدین ترتیب از تعداد کاندیدها کاسته خواهد شد. شکل ۹ نقشه‌ی عروق خونی شبکه‌ی (خروجی آستانه‌گذاری افقی) را نشان می‌دهد. سپس با حذف کاندیدهایی از شکل ۸ که بر روی عروق خونی قرار گرفته‌اند یک گام جلوتر می‌رویم. این خروجی در شکل ۱۰ نشان داده شده است.



شکل ۹) نقشه‌ی عروق خونی شبکیه (تصویر ورودی در سمت چپ و نقشه‌ی عروق در سمت راست)



شکل ۱۰) کاندیدهای باقی‌مانده بعد از اینکه کاندیدهای موجود بر روی رگ‌ها حذف شده‌اند.

در این مرحله ضایعات باقی‌مانده اندازه‌ی درستی ندارند بنابراین بر اساس روش ارائه‌شده در مقاله‌ی [۶] عمل رشد ناحیه را روی آن اعمال می‌کنیم. برای این کار باید ابتدا برای هر کاندید شدت سطح خاکستری پیکسل‌ها را محاسبه کرد. (سطح خاکستری $i\text{-green}$ را برای صفحه‌ی سبز $I\text{-green}$ و سطح خاکستری $i\text{-bg}$ را برای صفحه‌ی زمینه $I\text{-bg}$ به کار می‌بریم). صفحه‌ی زمینه را با اعمال یک فیلتر میانگین 25×25 به صفحه‌ی سبز بدست می‌آوریم. کمترین شدت خاکستری در $i\text{-green}$ را با $I\text{-darkest}$ نشان می‌دهیم. این مقادیر برای پیدا کردن مقدار آستانه‌ی t استفاده می‌شوند:

$$t = I_darkest - x(I_darkest - i_bg)$$

در این مقاله مقدار X برابر با ۰.۵ قرار داده شده است. سپس برای قطعه‌بندی I-green به یک تصویر باینری، از مقدار t استفاده خواهیم کرد. عملیات رشد ناحیه از پیکسلی با شدت سطح خاکستری به میزان I-darkest شروع شده و وقتی به پایان می‌رسد که پیکسل بیشتری به آن متصل نباشد. اگر ناحیه‌ی حاصل بزرگتر از ۱۲۰ پیکسل باشد به دلیل اینکه هیچ ضایعه‌ی قرمز رنگی نمی‌تواند به این بزرگی باشد از آن دست کشیده خواهد شد. شکل ۱۱ نتیجه‌ی حاصل از رشد ناحیه برای کاندیدهای موجود در شکل ۱۰ را نشان می‌دهد.



شکل ۱۱) نتیجه‌ی رشد ناحیه بر روی کاندیدهای موجود در شکل ۱۰

مرحله ۲: تشخیص ضایعات قرمز رنگ درست

در این مرحله تشخیص درست ضایعات قرمز رنگ از طریق استخراج ویژگی میسر می‌شود. در این مقاله برای هر کاندید بر اساس شکل، شدت سطح خاکستری پیکسل، شدت رنگ، پاسخ فیلترهای گوسی و ضریب همبستگی ۳۱ مشخصه مورد استفاده قرار گرفته است. در ادامه ۳۱ مشخصه‌ای که در این مقاله برای مشخص کردن ضایعات قرمز رنگ درست از آنها استفاده شده است را آورده ایم. مشخصه‌های ۲۷ تا ۲۹ خاص روش ارائه شده هستند چراکه براساس ضریب همبستگی‌ای می‌باشند که قبلاً برای هر پیکسل محاسبه شده است. دو

مشخصه‌ی نهایی، طول محور بزرگ و طول محور کوچک هستند که برای تنوع بخشیدن به مشخصه‌های شکلی موجود اضافه شده‌اند.

۱- پارامتر a (مساحت کاندید). ضایعات در مقایسه با سایر موارد موجود بر روی شبکه دارای مساحت کوچکتری هستند.

۲- پارامتر p (محیط کاندید). ضایعات در مقایسه با سایر موارد موجود بر روی شبکه دارای محیط کوچکتری هستند.

۳- پارامتر $r = l/w$ (نسبت طول محور بزرگ به محور کوچک کاندید). در ضایعات، طول محور بزرگ تقریباً با طول محور کوچک برابر است.

۴- پارامتر $c = 4\pi a/p^2$ (دایره‌ای شکل بودن کاندید). ضایعات عموماً دایره‌ای شکل هستند.

۵- پارامتر $i\text{-green}$ (شدت سطح خاکستری کاندید در صفحه‌ی سبز). ضایعات عموماً سطح خاکستری بالاتری دارند.

۶- پارامتر $i\text{-sc}$ (شدت سطح خاکستری کاندید در صفحه‌ی I-SC).

۷- میانگین شدت سطح خاکستری کاندید در صفحه‌ی سبز $(i\text{-green})$ و $m\text{-green}$ ($m\text{-green} = i\text{-green}/a$)

۸- میانگین شدت سطح خاکستری کاندید در صفحه‌ی سبز $(i\text{-sc})$ و $m\text{-sc}$ ($m\text{-sc} = i\text{-sc}/a$)

۹- پارامتر شدت نرمال شده در صفحه‌ی سبز، $NI_green = (1/\sigma)(i_green - x) \gg NI_green$ که σ در آن انحراف از معیار

و x در آن میانگین مقدار پیکسل‌ها در صفحه‌ی زمینه $I\text{-bg}$ می‌باشد).

۱۰- پارامتر شدت نرمال شده در صفحه‌ی I-SC، $NI_SC = (1/\sigma)(i_sc - x) \gg NI_SC$.

۱۱- پارامتر شدت میانگین نرمال شده در صفحه‌ی سبز، $NM_green = (1/\sigma)(m_green - x) \gg NM_green$.

۱۲- پارامتر شدت نرمال شده میانگین در صفحه‌ی I-SC، $NM_SC = (1/\sigma)(m_sc - x) \gg NM_SC$.

۱۳- پارامتر شدت $I\text{-darkest}$ در صفحه‌ی $I\text{-match}$

۱۴- پارامتر تراکم $v = \sqrt{\sum (d_i - d)^2 / n}$ که d_i در آن فاصله‌ی هر پیکسل مرزی از کاندید تا مرکز آن است. d میانگین

همه‌ی این فواصل و n تعداد پیکسل‌های مرزی می‌باشد. ضایعات عموماً فشرده هستند.

۱۵- اختلاف بین مقادیر میانگین پیکسل‌ها در کاندید و مقادیر میانگین پیکسل‌ها در صفحه‌ی قرمز در ناحیه‌ی دایره‌ای شکلی

که روی کاندید متمرکز شده‌است.

۱۶-۱۸ تکرار مشخصه‌ی ۱۵ اما این بار، عمل را بر روی کانال سبز، کانال آبی و کانال Hue از فضای رنگ HSI انجام می‌دهیم.

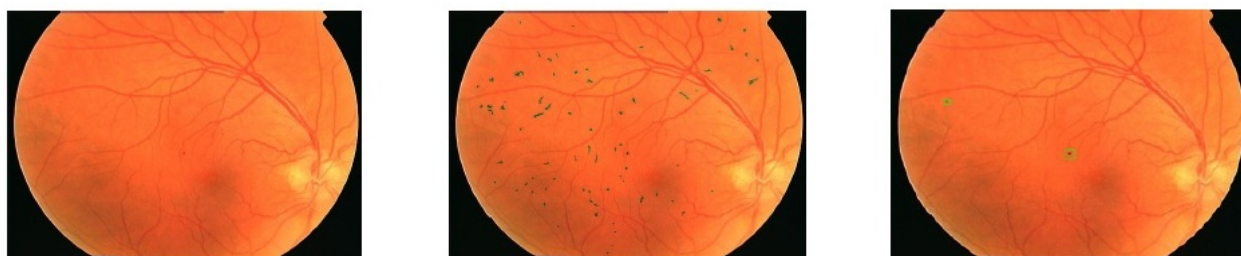
۱۹-۲۲ پاسخ فیلتر گوسی میانگین از صفحه‌ی سبز با سیگمای ۱، ۲، ۴ و ۸

۲۳-۲۶ پاسخ انحراف از معیار صفحه‌ی سبز، بعد از فیلترینگ گوسی با سیگمای ۱، ۲، ۴ و ۸

۲۷-۲۹ ضریب همبستگی مینیمم، ماکزیمم و میانگین از کاندید. ضایعات عموماً دارای ضریب همبستگی بالایی هستند.

۳۰- طول محور بزرگ کاندید.

۳۱- طول محور کوچک کاندید.



شکل ۱۲) خروجی تشخیص ضایعات قرمز رنگ (تصویر ورودی در سمت چپ- خروجی مرحله‌ی اول در وسط و خروجی

مرحله‌ی دوم در سمت راست)

بعد از اینکه مشخصه‌ها از کاندیدها استخراج شدند، آنها را به دو گروه تقسیم می‌کنیم: ضایعات درست و ضایعات غلط. سپس در گروه ضایعات درست از هر مشخصه، مقادیر ماکزیمم و مینیمم را بدست می‌آوریم و آنها را در یک جدول، به نام جدول تمایز ذخیره می‌کنیم. این جدول (جدول ۱)، برای حذف کاندیدهایی که ویژگی‌های آنها بالاتر از مقدار ماکسیمم و پایین‌تر از مقدار مینیمم است مورد استفاده قرار می‌گیرد. برای مشخصه‌ی ۱۶ که کنتراست بین کاندید و محیط اطرافش را در صفحه‌ی سبز مورد محاسبه قرار می‌دهد مقدار مینیمم را به صورت دستی برابر ۸ قرار داده‌ایم. با تنظیم دستی این مقدار، برخی ضایعات درست و خیلی از ضایعات غلط حذف می‌شوند. این مشخصه را به این دلیل انتخاب کرده‌ایم که کنتراست بین ضایعات و زمینه

اطرافشان بالا است. کاندیدهای باقی مانده‌ای که مقادیری بین مینیمم و ماکزیمم دارند، ضایعات قرمز رنگ نهایی تشخیص داده شده هستند.

Feature Number	Min.	Max.	Feature Number	Min.	Max.	Feature Number	Min.	Max.
1	2	109	12	0.164831	5.08576	23	36.00843	182.7637
2	2	113.0122	13	7	255	24	0.003416	3.00799
3	1	5.618986	14	0	3.743681	25	37.95791	183.7483
4	0.090521	6.283185	15	-31.368	2.863946	26	0.009462	1.797795
5	58	10676	16	8	35.37533	27	0.106603	0.785072
6	338	19451	17	-27.1471	6.368521	28	0	0.547187
7	25.17241	177.0417	18	-0.01944	0.002723	29	0.053335	0.658546
8	104.8125	221.9348	19	27.66061	179.0988	30	2.309401	29.53404
9	-0.31038	203.8908	20	0.01962	8.292589	31	1.154701	14.09998
10	6.138524	448.9355	21	30.81245	180.773			
11	-1.15234	0.921628	22	0.044188	6.747434			

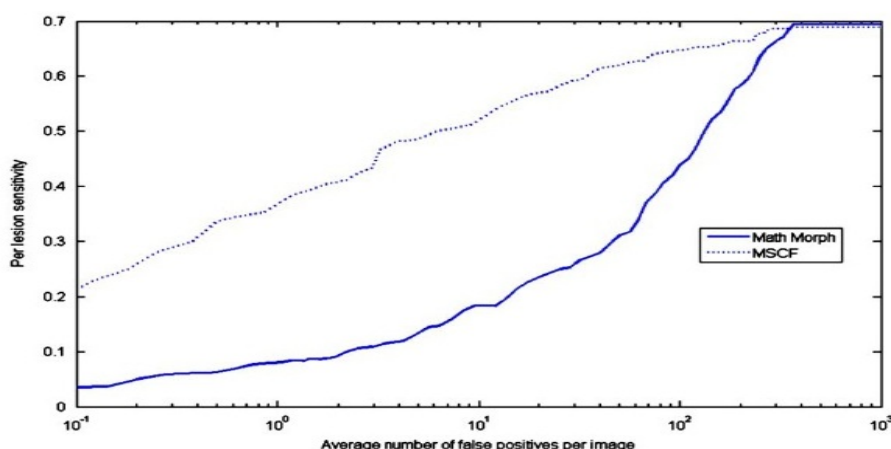
جدول (۱) جدول تمایز

نتایج بدست آمده از آزمایش‌ها و تحلیل آن‌ها:

برای ارزیابی الگوریتم پیشنهاد شده در این مقاله، بر روی ۱۰۰ تصویر شبکه‌ای از وبسایت ROC، توالی از آزمایش‌ها صورت گرفته است. این تصاویر با استفاده از دوربین‌های Topcon NW100، Canon NW200 و یا CR5-45NM با رزولوشن و تنظیمات فشرده‌سازی پیش‌فرض از بیمارانی که معلوم نیست دچار رتینوپاتی دیابتی هستند یا خیر تهیه شده است. همه‌ی تصاویر در فرمت فشرده‌شده‌ی JPEG هستند و دارای سایز ۷۶۸*۵۷۶، ۱۰۶۱*۱۰۵۸، ۱۰۶۱*۱۰۶۲، ۱۳۷۹*۱۳۸۳، ۱۳۸۱*۱۳۸۵، ۱۳۸۲*۱۳۸۵، ۱۳۸۴*۱۳۸۶، ۱۳۸۳*۱۳۸۹، ۱۳۹۱*۱۳۸۹ و ۱۳۹۲*۱۳۹۴ پیکسل می‌باشند. هر تصویر دارای یک مرجع استاندارد است که ضایعات قرمز رنگ در آن مشخص شده‌اند و مورد تایید چهار متخصص قرار گرفته است. این مرجع استاندارد برای تشخیص ضایعات قرمز رنگ صحیح در مرحله‌ی ساختن جدول تمایز به ما کمک خواهند کرد. نکته‌ای که باید در اینجا مد نظر داشت این است که برای قطعه‌بندی پاسخ نهایی بعد از اعمال هسته‌های گوسی، مقادیر

آستانه‌ای متغیر بین ۰.۱ تا ۱ بر روی تصاویر اعمال شده‌اند و هربار برحسب مقادیر مشخصه‌های مربوط به ضایعات قرمز رنگ درست، جدول تمایزی ساخته شده و برخی از کاندیدها حذف می‌شوند.

الگوریتم ارائه‌شده در این مقاله با الگوریتم Math Morph مورد مقایسه قرار گرفته‌است. در شکل ۱۴ کارایی رویکرد سلسله‌مراتبی ارائه‌شده در مقایسه با الگوریتم Math Morph به نمایش گذاشته شده‌است و این کار با رسم نمودار sensitivity برحسب تعداد میانگین false positive ها بر روی تصاویر test صورت گرفته‌است. منظور از sensitivity تعداد ضایعات درست تشخیص داده‌شده است در حالیکه false positive نشانگر تعداد مواردی است که به اشتباه ضایعه‌ی درست تشخیص داده‌شده‌اند. لازم به تذکر است که محور افقی در مقیاس لگاریتمی بیان شده‌است. همانطور که در شکل می‌توان دید الگوریتم سلسله‌مراتبی در اکثر موارد کارکرد بهتری را از خود نشان داده‌است.



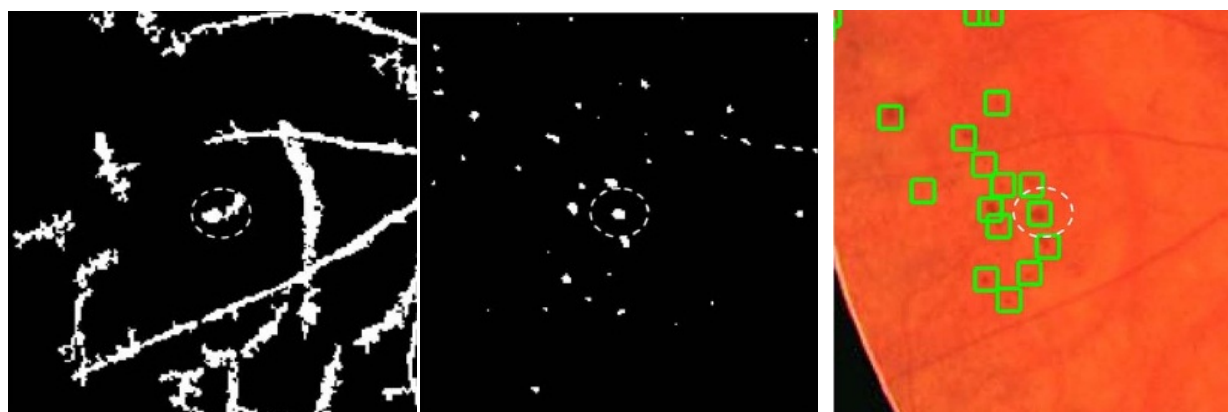
شکل ۱۴) مقایسه‌ی کارایی الگوریتم سلسله‌مراتبی با الگوریتم Math Morph بر روی داده‌های test

در جدول ۲ برای هر دو روش ذکر شده، sensitivity مربوط به false positive هایی با مقادیر ۰.۱، ۰.۲، ۰.۴، ۰.۸، ۰.۶، ۱، ۳.۲ و ۶.۴ به نمایش گذاشته شده‌است. این مقادیر از شکل ۱۴ بدست آمده‌اند و همانطور که مشاهده می‌شود روش سلسله‌مراتبی دارای sensitivity بالاتری می‌باشد.

False Positive Rate	MSCF	Math Morph
0.1	0.2296	0.0361
0.2	0.2810	0.0556
0.4	0.3361	0.0630
0.8	0.3744	0.0799
1.6	0.4081	0.0919
3.2	0.4808	0.1182
6.4	0.5071	0.1683
Average	0.3739	0.0876

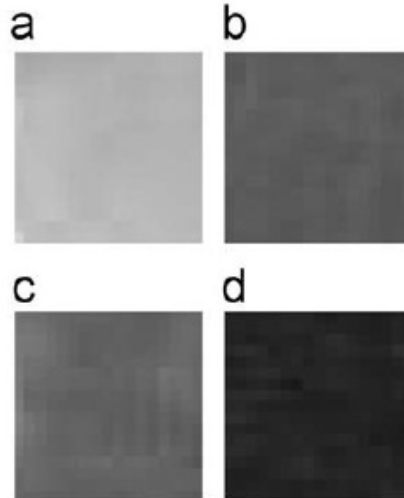
جدول ۲) sensitivity به ازای false positive های مختلف

تشخیص داده نشدن برخی از ضایعات درست قرمز رنگ می‌تواند به یکی از این دلایل باشد: ۱- کیفیت تصاویر ۲- انتخاب سیگمای نامناسب برای هسته‌ی گوسی ۳- نزدیک بودن بیش از اندازه‌ی ضایعه به شبکه‌ی عروق، ۴- کنتراست پایین بین ضایعات و زمینه‌ی اطراف آن. در شکل ۱۵ و ۱۶ برخی از این مشکلات به تصویر کشیده شده‌اند.



شکل ۱۵) سمت راست: ضایعات درست، وسط: کاندیدهای تشخیص داده‌شده، سمت چپ: تشخیص غلط ضایعه به عنوان

بخشی از شبکه‌ی عروق



شکل ۱۶) ضایعات درستی که دارای کنتراست پایینی با زمینه می‌باشند.

جمع‌بندی:

هدف کلی روش ارائه شده در این مقاله، تشخیص خودکار میکروآنوریسم‌ها در سطح شبکه می‌باشد. در این روش برای پیدا کردن اشکال مشکوک به میکروآنوریسم بر روی سطح شبکه از یک روش فیلترینگ چند مقیاسه بر مبنای همبستگی بیزین^{۱۱} استفاده شده است. برای بدست آوردن معیار همبستگی^{۱۲} به این صورت عمل شده است: ۱- با بکارگیری آرایه‌ای از فیلترهای میان گذر گوسی^{۱۳} مدلی احتمالاتی از یک شی و محیط اطراف آن ساخته می‌شود. ۲- با تطبیق خروجی فیلترها بر روی تصویر ورودی و مدل‌های آموزش دیده شده^{۱۴}، معیار همبستگی بدست می‌آید.

¹¹ Bayesian Correlation

¹² Correlation Measure

¹³ Gaussian Filter Bank

¹⁴ Train

وقتی پاسخ فیلترینگ همبستگی¹⁵ از یک مقدار آستانه بیشتر شد مکان‌های پیدا شده به عنوان مکانهای احتمالی میکروآنوریسم در نظر گرفته می‌شوند. از آنجایی که میکروآنوریسم‌ها نمی‌توانند بر روی عروق خونی رخ بدهند، در مرحله‌ی بعد به کمک یک شمای افقی ابتدا نقشه‌ی شبکه‌ی رگها در چشم مشخص می‌شود سپس با کم کردن خروجی مرحله‌ی قبل و خروجی این مرحله مکانهای غلط مشخص شده بر روی رگها حذف می‌شوند. در مرحله‌ی بعد میکروآنوریسم‌های کاندید شده با استفاده از روش رشد دادن ناحیه¹⁶ قطعه بندی می‌شوند و مجموعه‌ی بزرگی از ویژگی‌ها برپایه‌ی شکل، شدت سطح خاکستری پیکسل‌ها، پاسخ فیلترهای میان‌گذر گوسی و مقدار ضریب همبستگی از هر کاندید استخراج می‌شوند. سپس مقدار مینیمم و ماکزیمم هر ویژگی برای تمام ضایعه‌ها محاسبه شده و در یک جدول به نام جدول تمایز¹⁷ قرار داده می‌شوند. این جدول برای حذف کاندیدهایی که ویژگی‌های آنها بالاتر از مقدار ماکسیمم و پایین‌تر از مقدار مینیمم است مورد استفاده قرار می‌گیرد. بعد از این مرحله کاندیدهای باقی‌مانده به عنوان ضایعات قرمز رنگ درست طبقه‌بندی خواهند شد.

¹⁵ Correlation Filtering

¹⁶ Region Growing

¹⁷ Discrimination Table

فصل ۲: گزارش پیاده‌سازی مقاله و نتایج بدست آمده

مقدمه:

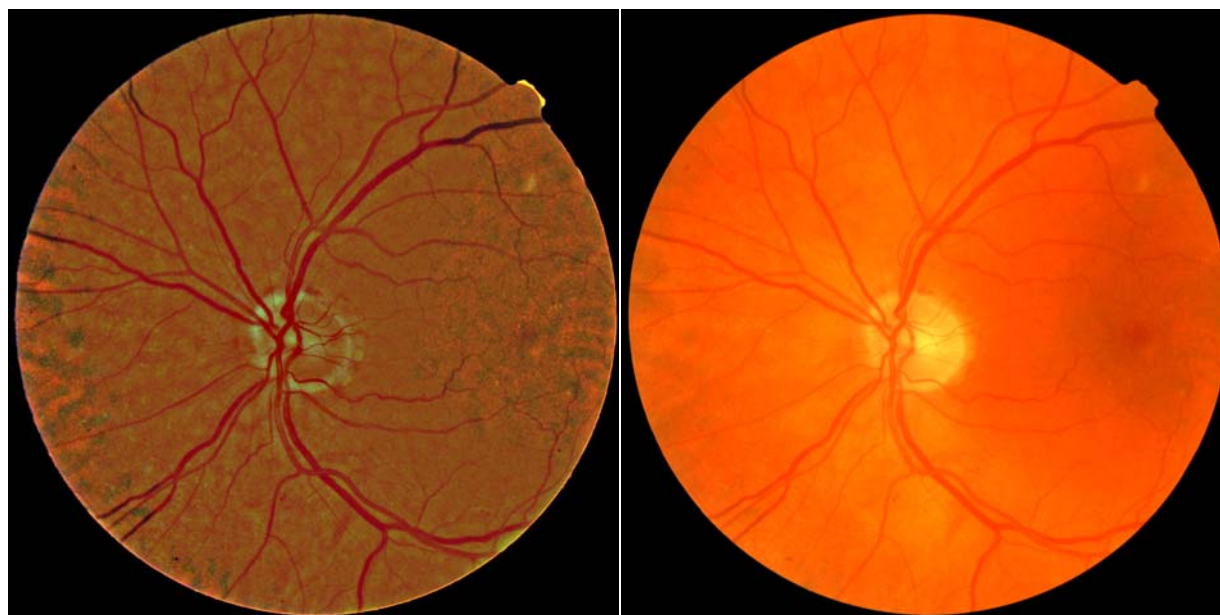
در این بخش گزارشی از پیاده‌سازی مقاله ارائه شده‌است. در قسمت نخست پیاده‌سازی، ابتدا شبکه‌ی عروق را در شبکه بدست آورده‌ایم. این کار را ابتدا با پیاده‌سازی مقاله‌ی [۷] و با بکارگیری الگوهای Kirsch انجام دادیم (کد مربوط به پیاده‌سازی در بخش ۱ آورده شده‌است) اما چون کیفیت رگ‌های بدست آمده در آن وابسته به شرایط تصویر و آستانه‌ی انتخاب شده بود از الگوریتم دیگری کمک گرفتیم که کد متلب مربوط به آن در بخش ۲ آورده شده‌است. کیفیت رگ‌های بدست آمده از این الگوریتم که بر مبنای تبدیل رادون می‌باشد به مراتب بهتر از الگوریتم اول است اما متأسفانه اجرای آن بسیار زمان‌بر است.

در قسمت بعد، فاز اول الگوریتم تشخیص سلسله‌مراتبی میکروآنوریسم‌ها پیاده‌سازی شده‌است. این فاز مربوط به تشخیص میکروآنوریسم‌های کاندید می‌باشد و خود شامل چندین زیربخش است: ۱- ساختن مدلی احتمالاتی از یک شی و محیط اطراف آن با بکارگیری آرایه‌ای از فیلترهای میان گذر گوسی ۲- بدست آوردن ضریب همبستگی با تطبیق خروجی فیلترها بر روی تصویر ورودی ۳- بدست آوردن میکروآنوریسم‌های کاندید

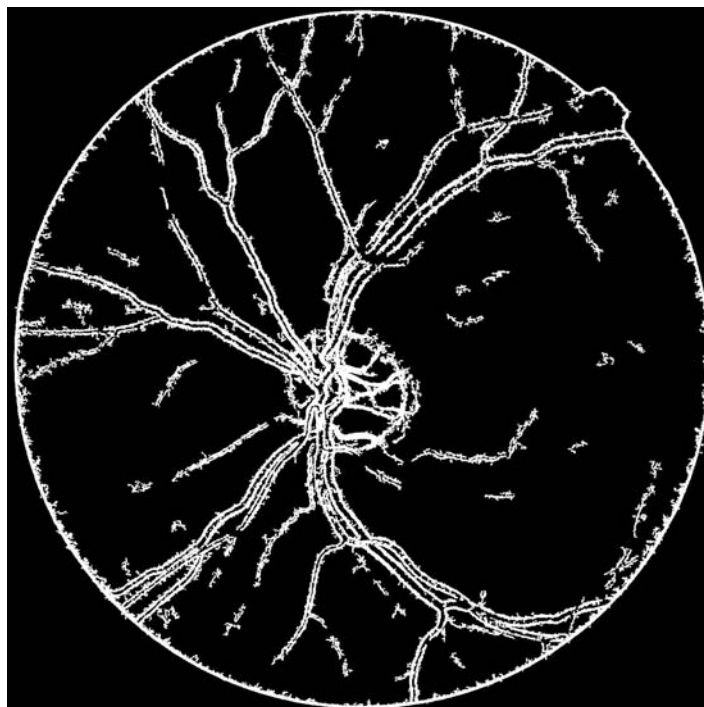
با در نظر گرفتن مکان‌هایی که پاسخ فیلترینگ همبستگی آن‌ها از یک مقدار آستانه بیشتر شده‌است. ۴- حذف کاندیدهایی که بر روی عروق خونی مشخص شده‌اند. ۵- قطعه‌بندی میکروآنوریسم‌های کاندید شده با استفاده از روش رشد ناحیه. کد مربوط به فاز اول در بخش ۳ قابل مشاهده می‌باشد.

در قسمت نهایی، فاز دوم الگوریتم پیاده‌سازی شده‌است. این فاز مربوط به تشخیص میکروآنوریسم‌های درست می‌باشد و شامل این زیربخش‌ها می‌شود: ۱- استخراج مجموعه‌ی بزرگی از ویژگی‌ها از هر کاندید ۲- تشکیل جدول تمایز بر پایه‌ی مقدار مینیمم و ماکزیمم هر ویژگی ۳- حذف کاندیدهایی که ویژگی‌های آن‌ها بالاتر از مقدار ماکزیمم و پایین‌تر از مقدار مینیمم است. کد مربوط به فاز دوم در بخش ۴ قابل مشاهده می‌باشد.

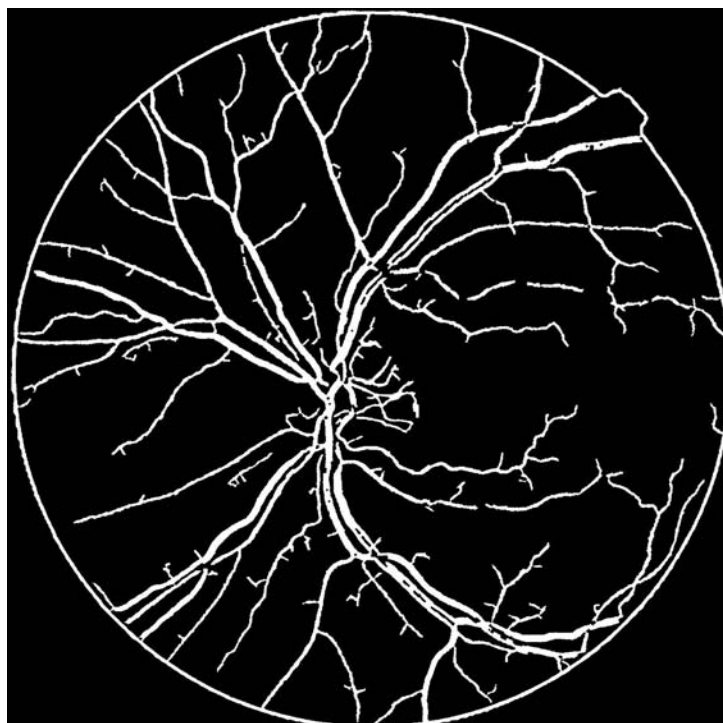
نتایج:



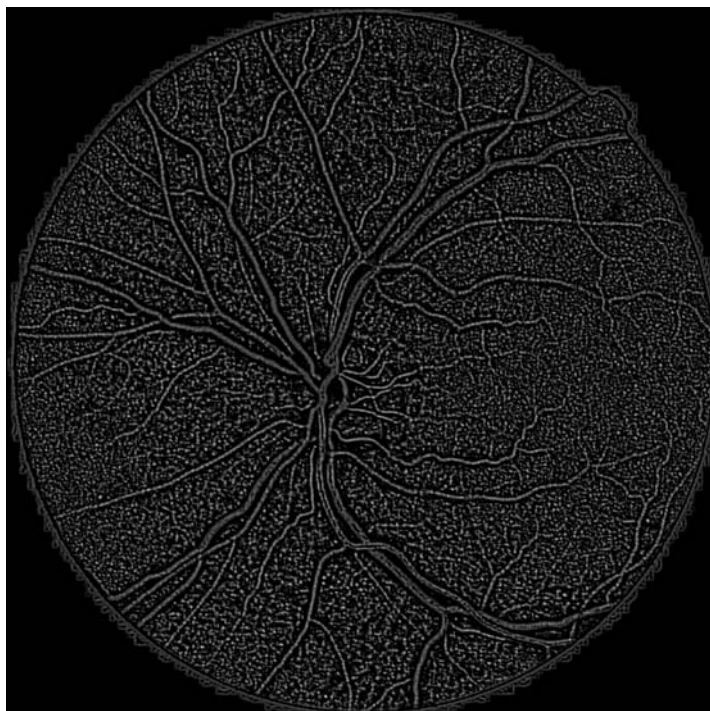
شکل ۱: تصویر ورودی و نرمالیزه شده‌ی رنگ در آن



شکل ۲: استخراج شبکه‌ی عروق به کمک الگوهای Kirsch



شکل ۳: استخراج شبکه‌ی عروق به کمک تبدیل رادون



شکل ۴: بدست آوردن ضریب همبستگی ماکزیمم با تطبیق خروجی فیلترها بر روی تصویر ورودی



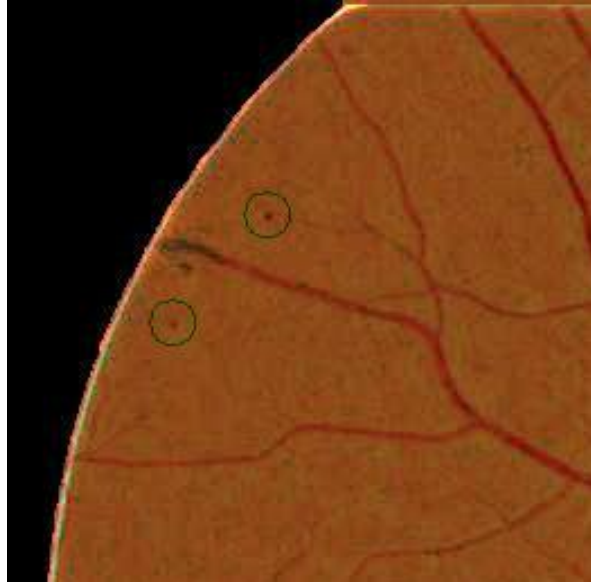
شکل ۵: اعمال آستانه بر روی خروجی مرحله ی قبل



شکل ۶: حذف کاندیدهایی که بر روی شبکه‌ی عروق قرار گرفته‌اند



شکل ۷: خروجی عملیات رشد ناحیه



شکل ۸: تشخیص صحیح میکروآنوریزمها

پیوست:

بخش ۱: استخراج شبکه‌ی عروق با بکارگیری الگوهای Kirsch

```
%-----
% Programmer: Bahareh Bafandeh
%-----

function bloodVessels = VesselExtract(inImg, threshold)
[r c]=size(inImg);
bloodVessels=zeros(r,c);
%Kirsch's Templates
h1=[5 -3 -3; 5 0 -3; 5 -3 -3]/15;
h2=[-3 -3 5; -3 0 5; -3 -3 5]/15;
h3=[-3 -3 -3; 5 0 -3; 5 5 -3]/15;
h4=[-3 5 5; -3 0 5; -3 -3 -3]/15;
h5=[-3 -3 -3; -3 0 -3; 5 5 5]/15;
h6=[ 5 5 5; -3 0 -3; -3 -3 -3]/15;
h7=[-3 -3 -3; -3 0 5; -3 5 5]/15;
h8=[ 5 5 -3; 5 0 -3; -3 -3 -3]/15;

%Spatial Filtering
t1=filter2(h1,inImg);
t2=filter2(h2,inImg);
t3=filter2(h3,inImg);
t4=filter2(h4,inImg);
t5=filter2(h5,inImg);
t6=filter2(h6,inImg);
t7=filter2(h7,inImg);
t8=filter2(h8,inImg);

bV = max(t1,t2);
bV = max(bV,t3);
```

```

bV = max(bV,t4);
bV = max(bV,t5);
bV = max(bV,t6);
bV = max(bV,t7);
bV = max(bV,t8);

for i=1:r
    for j=1:c
        if(bV(i,j)>threshold)
            bloodVessels(i,j)=bV(i,j);
        end
    end
end
end

```

بخش ۲: استخراج شبکه‌ی عروق با کمک تبدیل رادون

```

clc;clear;close all
warning off
global BlkMsk TetaStp;

%Parameters
%   n=13;
%   stp=3;
%   n=13;
%   LinVldThr = 3.3;
%   stp=3;
%   TetaStp = 4;
Trans.
%   LinVldThr = 3.3;
%End Parameters

% Circular Mask
BlkMsk = ones(n,n);
nh = fix(n/2)+1;
rad2 = nh^2;
for r = 1:n
    for c = 1:n
        if (r-nh)^2+(c-nh)^2 > rad2
            BlkMsk(r,c) = 0;
        end
    end
end

for ii=0:0
f=imread(strcat('images\image',int2str(ii),'_test.jpg'));
%f=imresize(f,[480 640]);
[R,C,D]=size(f);
msk=zeros(R, C, 'uint8');
vesselWidth=zeros(R, C, n, 'uint8');

```

```

for r=1:fix(n/stp):R-n+1
    for c=1:fix(n/stp):C-n+1
        fl=f(r:r+n-1,c:c+n-1,:);
        [mx LineStrt LineEnd LineAngle LineVld]=LocRadFun1(fl);
        if LineVld>(LinVldThr/n)
            ml=zeros(n,n); mz=zeros(n,n);
            ml(:,LineStrt:LineEnd)=1;
            ml=[mz ml mz; mz ml mz; mz ml mz];
            ml=imrotate(ml,LineAngle,'crop');
            mskl=uint8(ml(n+1:2*n,n+1:2*n));
            f2=fl(:, :, 2);    f2=f2.*mskl;    t1=mean(f2(f2~=0));
            f2=fl(:, :, 2);    f2=f2.*(1-mskl);    t2=mean(f2(f2~=0));
            f2=fl(:, :, 2);    f2=uint8(im2bw(im2double(f2),
(t1+t2)/2/255));
            mskl=mskl.*(1-f2);
            msk(r:r+n-1,c:c+n-1)=msk(r:r+n-1,c:c+n-1)|mskl;
        end
    end
end

g1=255-f(:, :, 2);
imwrite(g1, strcat('graytest\graytest',int2str(ii),'.jpg'));
    %figure(2); imshow(g1);

    %figure(3); imshow(msk*255);
imwrite(msk*255, strcat('masktest\masktest',int2str(ii),'.jpg'));
msk=imclose(msk,strel('disk', 2));
%    figure(4); imshow(msk*255);
imwrite(msk*255, strcat('modmasktest\modmasktest',int2str(ii),'.jpg'));

em=edge(msk);    em=1-em;
imwrite(em, strcat('edgtest\edgtest',int2str(ii),'.jpg'));
%    figure(5); imshow(em);

    %f(:, :, 1)=f(:, :, 1).*em;
f(:, :, 1)=f(:, :, 1).*uint8(em);    f(:, :, 3)=f(:, :, 3).*uint8(em);
%    figure(6); imshow(f);
imwrite(f, strcat('resulttest\resulttest',int2str(ii),'.jpg'));
end

function [mx LStrt LEnd LineAngle LineVld]=locradfun1(f)
global BlkMsk TetaStp;
%figure(1); imshow(f(:, :, 2));
Level=0.7;    %#####Line Width Control

[R C D]=size(f);
f=double(f);
g1=255-f(:, :, 2);
g1=g1/255;    g1=imadjust(g1);
immean = mean2(g1);
g1=g1.*BlkMsk;
%figure(9); imshow(g1);

theta = 0:TetaStp:180;

```

```

[Rd,xp] = radon(g1,theta);

h=fspecial('average',3);
Rd=imfilter(Rd,h);

[mv mi]=max(Rd);
[mx mc]=max(mv);
mr=mi(mc);

LineAngle=mc*TetaStp;

if LineAngle > 90
    OrtAngle=fix((LineAngle-90)/TetaStp);
else
    OrtAngle=fix((LineAngle+90)/TetaStp);
end

maxcul=Rd(:,mc);
maxcul=maxcul-R*immean;      maxcul=maxcul.*((sign(maxcul)+1)/2);
%figure(2); plot(maxcul);
[mx LinePos]=max(maxcul);

if mx == 0
    LineVld=0;
    LStrt=0;
    LEnd=0;
else
    %LinePos=mr;
    LineVld=mx/R;
    [RR RC]=size(Rd);

    LStrt=LinePos;
    LEnd=LinePos;
    for i=LinePos:-1:1
        LStrt=i;
        if maxcul(i)<Level*mx
            break
        end
    end
    for i=LinePos:1:RC
        LEnd=i;
        if maxcul(i)<Level*mx
            break
        end
    end

    LStrt=LStrt-fix(RR/2)+fix(R/2);
    LEnd=LEnd-fix(RR/2)+fix(R/2);
    %[LStrt LEnd]
    if LStrt<1
        LStrt=1;
    end
    if LEnd>R
        LEnd=R;
    end
end

```

```

    %[LStrt LEnd]
end

```

بخش ۳: کد مربوط به فاز اول (تشخیص میکروآنوریسم‌های کاندید)

```

function
[Stagel,Maxrr,Minrr,Avgrrr]=CandidateMicroExtract(NormInput,bloodVessels,input
)
[R C]=size(input);

gussian1=zeros(9,9);
gussian2=zeros(9,9);
gussian3=zeros(9,9);
gussian4=zeros(11,11);
gussian5=zeros(11,11);

sigma1=1.1;
sigma2=1.2;
sigma3=1.3;
sigma4=1.4;
sigma5=1.5;

for i=1:9
    for j=1:9
        gussian1(i,j)=(1/(2*pi*(sigma1^2)))*exp(-((i-5)^2+(j-
5)^2)/(2*(sigma1^2)));
    end
end

for i=1:9
    for j=1:9
        gussian2(i,j)=(1/(2*pi*(sigma2^2)))*exp(-((i-5)^2+(j-
5)^2)/(2*(sigma2^2)));
    end
end

for i=1:9
    for j=1:9
        gussian3(i,j)=(1/(2*pi*(sigma3^2)))*exp(-((i-5)^2+(j-
5)^2)/(2*(sigma3^2)));
    end
end
%%%
for i=1:11
    for j=1:11
        gussian4(i,j)=(1/(2*pi*(sigma4^2)))*exp(-((i-6)^2+(j-
6)^2)/(2*(sigma4^2)));
    end
end
%%%
for i=1:11
    for j=1:11
        gussian5(i,j)=(1/(2*pi*(sigma5^2)))*exp(-((i-6)^2+(j-
6)^2)/(2*(sigma5^2)));
    end
end

```

```

    end
end

gussian1=imcomplement((mat2gray(gussian1)));
gussian2=imcomplement((mat2gray(gussian2)));
gussian3=imcomplement((mat2gray(gussian3)));
gussian4=imcomplement((mat2gray(gussian4)));
gussian5=imcomplement((mat2gray(gussian5)));

Maxrr=zeros(R,C);
Minrr=zeros(R,C);
Avgrr=zeros(R,C);

r1=CorrAB(NormInput,gussian1);
r2=CorrAB(NormInput,gussian2);
r3=CorrAB(NormInput,gussian3);
r4=CorrAB(NormInput,gussian4);
r5=CorrAB(NormInput,gussian5);

for i=5:R-6
    for j=5:C-6
        r=i-4;
        c=j-4;
        A=[r1(r,c),r2(r,c),r3(r,c),r4(r,c),r5(r,c)];
        Maxrr(i,j)=max(A);
        Minrr(i,j)=min(A);
        Avgrr(i,j)=mean(A);
    end
end
imwrite(Maxrr,'Maxrr.jpg');

%threshold1
output=(Maxrr>0.4);
imwrite(output,'output.jpg');

%Removing any candidates on the vessels
Stagel=output-bloodVessels;
for r=1:R
    for c=1:C
        if Stagel(r,c)<0
            Stagel(r,c)=0;
        end
    end
end
imwrite(Stagel,'Stagel.jpg');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Region Growing %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mask1 = false(size(input));
[L1,num1]=bwlabel(Stagel);
stats1 = regionprops(L1,'all');

for i=1:num1
    vlcent=stats1(i).Centroid;

```



```

jm=vlcent(1,1);
im=vlcent(1,2);
rms=2;
[x,y]=meshgrid((im-rms):(im+rms),(jm-rms):(jm+rms));
c_mask=((x-im).^2+(y-jm).^2)<=rms^2;
[r1 c1]=size(c_mask);
for cnt1=1:r1
    for cnt2=1:c1
        if c_mask(cnt1,cnt2)==1
            x1=round(x(cnt1,cnt2));
            y1=round(y(cnt1,cnt2));
            if x1>0 && x1<R && y1>0 && y1<C
                mask1(x1,y1)=1;
            end
        end
    end
end
end
Stagel=mask1-bloodVessels;
imwrite(Stagel,'RegionGrowing.jpg');

```

بخش ۴: کد مربوط به فاز دوم (تشخیص میکروآنورسم‌های صحیح)

```

function ind=PossibleLesions(minn,maxx)
%-----
% Programmer: Bahareh Bafandeh
%-----
load('FeatureFile.mat');

ind1=(a>=minn(1) & a<=maxx(1));
ind2=(per>=minn(2) & per<=maxx(2));
ind3=(MrM>=minn(3) & MrM<=maxx(3));
ind4=(circularity>=minn(4) & circularity<=maxx(4));
ind5=(i_green>=minn(5) & i_green<=maxx(5));
ind6=(i_sc>=minn(6) & i_sc<=maxx(6));
ind7=(m_green>=minn(7) & m_green<=maxx(7));
ind8=(m_sc>=minn(8) & m_sc<=maxx(8));
ind9=(NI_green>=minn(9) & NI_green<=maxx(9));
ind10=(NI_sc>=minn(10) & NI_sc<=maxx(10));
ind11=(NM_green>=minn(11) & NM_green<=maxx(11));
ind12=(NM_sc>=minn(12) & NM_sc<=maxx(12));
ind13=(IDarkest>=minn(13) & IDarkest<=maxx(13));
ind14=(FiltSig1>=minn(14) & FiltSig1<=maxx(14));
ind15=(FiltSig2>=minn(15) & FiltSig2<=maxx(15));
ind16=(FiltSig4>=minn(16) & FiltSig4<=maxx(16));
ind17=(FiltSig8>=minn(17) & FiltSig8<=maxx(17));
ind18=(StdFiltSig1>=minn(18) & StdFiltSig1<=maxx(18));
ind19=(StdFiltSig2>=minn(19) & StdFiltSig2<=maxx(19));
ind20=(StdFiltSig4>=minn(20) & StdFiltSig4<=maxx(20));
ind21=(StdFiltSig8>=minn(21) & StdFiltSig8<=maxx(21));
ind22=(MaxCorr>=minn(22) & MaxCorr<=maxx(22));
ind23=(MinCorr>=minn(23) & MinCorr<=maxx(23));
ind24=(AvgCorr>=minn(24) & AvgCorr<=maxx(24));
ind26=(diffg>=minn(26) & diffg<=maxx(26));
ind27=(diffg>=minn(27) & diffg<=maxx(27));
ind28=(diffb>=minn(28) & diffb<=maxx(28));

```

```

ind29=(diffh>=minn(29) & diffh<=maxx(29));
ind30=(MajorAxis>=minn(30) & MajorAxis<=maxx(30));
ind31=(MinorAxis>=minn(31) & MinorAxis<=maxx(31));

indd=ind1&ind2&ind3&ind4&ind5&ind6&ind7&ind8&ind9&ind10&ind11&ind12&ind13&ind
14&ind15&ind16&ind17;
ind=indd&ind18&ind19&ind20&ind21&ind22&ind23&ind24&ind26&ind27&ind28&ind29&in
d30&ind31;

```

بخش 5: کد مربوط به بخش اصلی (مرحله ی training)

```

%-----
% Programmer: Bahareh Bafandeh
%-----
clc;
clear all;
countertable=1;
xDoc = xmlread('annotations-consensus-ma-only.xml');
strc = xml2struct( xDoc );
minn=zeros(1,31);
maxx=zeros(1,31);
minn(1:31)=100000;
maxx(1:31)=-100000;

for jj=0:49
d=strc.set.annotations_dash_per_dash_image(jj+1);
d=d{1,1};

if isfield(d,'annotation')

inputt=double(imread(strcat('training\image',int2str(jj),'_training.jpg')));
input= inputt(:, :, 2);

input_img=imread(strcat('normalization\normalization',int2str(jj),'.jpg'));
input1=double(input_img);
NormInput= input1(:, :, 2);

[R C]=size(input);
bloodVessels=mat2gray(imread(strcat('modmask\modmask',int2str(jj),'.jpg')));

[output4,Maxrr,Minrr,Avgrr]=CandidateMicroExtract(NormInput,bloodVessels,inp
ut);
[stats,co]=FeatureExtract(input1,output4,input,Maxrr,Minrr,Avgrr);

load('FeatureFile.mat');

[sz ~]=size(a);
if sz~=0
    minn(1)=min(min(a),minn(1));
    maxx(1)=max(max(a),maxx(1));
    minn(2)=min(min(per),minn(2));
    maxx(2)=max(max(per),maxx(2));
    minn(3)=min(min(MrM),minn(3));
    maxx(3)=max(max(MrM),maxx(3));
    minn(4)=min(min(circularity),minn(4));
    maxx(4)=max(max(circularity),maxx(4));

```

```

minn(5)=min(min(i_green),minn(5));
maxx(5)=max(max(i_green),maxx(5));
minn(6)=min(min(i_sc),minn(6));
maxx(6)=max(max(i_sc),maxx(6));
minn(7)=min(min(m_green),minn(7));
maxx(7)=max(max(m_green),maxx(7));
minn(8)=min(min(m_sc),minn(8));
maxx(8)=max(max(m_sc),maxx(8));
minn(9)=min(min(NI_green),minn(9));
maxx(9)=max(max(NI_green),maxx(9));
minn(10)=min(min(NI_sc),minn(10));
maxx(10)=max(max(NI_sc),maxx(10));
minn(11)=min(min(NM_green),minn(11));
maxx(11)=max(max(NM_green),maxx(11));
minn(12)=min(min(NM_sc),minn(12));
maxx(12)=max(max(NM_sc),maxx(12));
minn(13)=min(min(IDarkest),minn(13));
maxx(13)=max(max(IDarkest),maxx(13));
minn(14)=min(min(FiltSig1),minn(14));
maxx(14)=max(max(FiltSig1),maxx(14));
minn(15)=min(min(FiltSig2),minn(15));
maxx(15)=max(max(FiltSig2),maxx(15));
minn(16)=min(min(FiltSig4),minn(16));
maxx(16)=max(max(FiltSig4),maxx(16));
minn(17)=min(min(FiltSig8),minn(17));
maxx(17)=max(max(FiltSig8),maxx(17));
minn(18)=min(min(StdFiltSig1),minn(18));
maxx(18)=max(max(StdFiltSig1),maxx(18));
minn(19)=min(min(StdFiltSig2),minn(19));
maxx(19)=max(max(StdFiltSig2),maxx(19));
minn(20)=min(min(StdFiltSig4),minn(20));
maxx(20)=max(max(StdFiltSig4),maxx(20));
minn(21)=min(min(StdFiltSig8),minn(21));
maxx(21)=max(max(StdFiltSig8),maxx(21));
minn(22)=min(min(MaxCorr),minn(22));
maxx(22)=max(max(MaxCorr),maxx(22));
minn(23)=min(min(MinCorr),minn(23));
maxx(23)=max(max(MinCorr),maxx(23));
minn(24)=min(min(AvgCorr),minn(24));
maxx(24)=max(max(AvgCorr),maxx(24));
minn(26)=min(min(diffrr),minn(26));
maxx(26)=max(max(diffrr),maxx(26));
minn(27)=min(min(diffg),minn(27));
maxx(27)=max(max(diffg),maxx(27));
minn(28)=min(min(diffb),minn(28));
maxx(28)=max(max(diffb),maxx(28));
minn(29)=min(min(diffh),minn(29));
maxx(29)=max(max(diffh),maxx(29));
minn(30)=min(min(MajorAxis),minn(30));
maxx(30)=max(max(MajorAxis),maxx(30));
minn(31)= min(min(MinorAxis),minn(31));
maxx(31)= max(max(MinorAxis),maxx(31));
end
end
end
save('minmax4file.mat','minn','maxx');

```

بخش ۶: کد مربوط به بخش اصلی (مرحله ی test)

```
%-----
% Programmer: Bahareh Bafandeh
%-----

clc;clear all;close all;

for jj=3:3
inputt=double(imread(strcat('training\image',int2str(jj),'_training.jpg')));
input= inputt(:, :, 2);

input_img=imread(strcat('normalization\normalization',int2str(jj),'.jpg'));
input1=double(input_img);
NormInput= input1(:, :, 2);

[R C]=size(input);
bloodVessels=mat2gray(imread(strcat('modmask\modmask',int2str(jj),'.jpg')));

[output4,Maxrr,Minrr,Avgr]=CandidateMicroExtract(NormInput,bloodVessels,input);
[stats,co]=FeatureExtract(input1,output4,input,Maxrr,Minrr,Avgr);

load('minmax1file.mat','minn1','maxx1');
load('minmax2file.mat','minn2','maxx2');
load('minmax4file.mat','minn4','maxx4');
load('minmax7file.mat','minn7','maxx7');

[ind1]=PossibleLesions(minn1,maxx1);
[ind2]=PossibleLesions(minn2,maxx2);
[ind4]=PossibleLesions(minn4,maxx4);
[ind7]=PossibleLesions(minn7,maxx7);
ind=ind1&ind2&ind4&ind7;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

finalmask = false(size(input));

for i=1:co
    if ind(i)==1
        vlcent=stats(i).Centroid;
        im=vlcent(1,1);
        jm=vlcent(1,2);
        rms=10;
        [x,y]=meshgrid((jm-rms):(jm+rms),(im-rms):(im+rms));
        c_mask=((y-im).^2+(x-jm).^2)<=rms^2);
        [r1 c1]=size(c_mask);
        for cnt1=1:r1
            for cnt2=1:c1
                if c_mask(cnt1,cnt2)==1
                    x1=round(x(cnt1,cnt2));
                    y1=round(y(cnt1,cnt2));
                    if x1>0 && x1<R && y1>0 && y1<C
                        finalmask(x1,y1)=1;
                    end
                end
            end
        end
    end
end
```

```

        end
    end
end
imshow(finalmask);

em=edge(finalmask); em=1-em;
input_img(:,:,1)=input_img(:,:,1).*uint8(em);
input_img(:,:,3)=input_img(:,:,3).*uint8(em);
imwrite(input_img, strcat('FinalResultTest\FinalResultTest',int2str(jj),'.jpg'
));
end

```

بخش ۷: کد مربوط به توابع استفاده شده

تابع FeatureExtract

```

function [stats,co]=FeatureExtract(input1,output4,input,Maxrr,Minrr,Avgrr)
%-----
% Programmer: Bahareh Bafandeh
%-----
[R C]=size(input);
Background = medfilt2(input, [25 25]);

% % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Feature Extraction %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[L,co] = bwlabel(output4);
stats = regionprops(L, 'all');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Feature 1 (a)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
a = [stats.Area];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Feature 2 (p)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
per = [stats.Perimeter];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Feature 3 (r),4 (c),5 (i_green),7 (m_green),8 (m_sc)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 9 (NI_green),10 (NI_SC),11 (NM_green),12 (NM_sc),13 (I-darkest)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 19 (FiltSig1),20 (FiltSig2),21 (FiltSig4),22 (FiltSig8)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Feature 23 (StdFiltSig1),Feature 24 (StdFiltSig2)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Feature 25 (StdFiltSig4),Feature 26 (StdFiltSig8)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Feature 27 (MaxCorr),Feature 28 (MinCorr),Feature 29 (AvgCorr)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MrM=zeros(1,co);
circularity=zeros(1,co);
i_green=zeros(1,co);
SC=input-Background;
i_sc=zeros(1,co);
m_green=zeros(1,co);
m_sc=zeros(1,co);
NI_green=zeros(1,co);
p = Background';
b = p(:)';
NI_sc=zeros(1,co);
Sig=std(b,1);
Mean=mean(b);
NM_green=zeros(1,co);

```

```

NM_sc=zeros(1,co);
IDarkest=zeros(1,co);
FiltSig1=zeros(1,co);
h = fspecial('gaussian',[7,7], 1);
gaus1= imfilter(input,h);
FiltSig2=zeros(1,co);
h = fspecial('gaussian',[13,13], 2);
gaus2= imfilter(input,h);
FiltSig4=zeros(1,co);
h = fspecial('gaussian',[25,25], 4);
gaus4= imfilter(input,h);
FiltSig8=zeros(1,co);
h = fspecial('gaussian',[33,33], 8);
gaus8= imfilter(input,h);
StdFiltSig1=zeros(1,co);
h = fspecial('gaussian',[7,7], 1);
gaus11= imfilter(input,h);
StdFiltSig2=zeros(1,co);
h = fspecial('gaussian',[13,13], 2);
gaus22= imfilter(input,h);
StdFiltSig4=zeros(1,co);
h = fspecial('gaussian',[25,25], 4);
gaus44= imfilter(input,h);
StdFiltSig8=zeros(1,co);
h = fspecial('gaussian',[33,33], 8);
gaus88= imfilter(input,h);
MaxCorr=zeros(1,co);
MinCorr=zeros(1,co);
AvgCorr=zeros(1,co);

for i=1:co
    MrM(1,i)=stats(i).MajorAxisLength/stats(i).MinorAxisLength;
    circularity(1,i)=(4*pi*a(1,i))/(per(1,i)^2);
    vl=L==i;
    i_green(i)=sum(sum(input(vl)));
    i_sc(i)=sum(sum(SC(vl)));
    m_green(1,i)=i_green(1,i)/a(1,i);
    m_sc(1,i)=i_sc(1,i)/a(1,i);
    NI_green(1,i)=(1/Sig)*(i_green(1,i)-Mean);
    NI_sc(1,i)=(1/Sig)*(i_sc(1,i)-Mean);
    NM_green(1,i)=(1/Sig)*(m_green(1,i)-Mean);
    NM_sc(1,i)=(1/Sig)*(m_sc(1,i)-Mean);
    IDarkest(i)=min(min(input(vl)));
    sums=sum(sum(vl));
    FiltSig1(i) = sum(sum(gaus1(vl)))/sums;
    FiltSig2(i) = sum(sum(gaus2(vl)))/sums;
    FiltSig4(i) = sum(sum(gaus4(vl)))/sums;
    FiltSig8(i) = sum(sum(gaus8(vl)))/sums;
    %[r0 c0]=size(vl);
    n1 = gaus11(vl);
    non0 = nonzeros(n1);
    StdFiltSig1(1,i)=std(non0,1);
    n1 = gaus22(vl);
    non0 = nonzeros(n1);
    StdFiltSig2(1,i)=std(non0,1);

```

```

n1 = gaus44(v1);
non0 = nonzeros(n1);
StdFiltSig4(1,i)=std(non0,1);
n1 = gaus88(v1);
non0 = nonzeros(n1);
StdFiltSig8(1,i)=std(non0,1);
MaxCorr(1,i)=sum(sum(Maxrr(v1)))/sums;
MinCorr(1,i)=sum(sum(Minrr(v1)))/sums;
AvgCorr(1,i)=sum(sum(Avgrr(v1)))/sums;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Feature 15,16,17,18
(diff)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

hsv = rgb2hsv(input1);
Hueinput=hsv(:, :, 1);
Redinput= input1(:, :, 1);
Greeninput= input1(:, :, 2);
Blueinput= input1(:, :, 3);

mm_green=zeros(1,co);
diffr=zeros(1,co);
diffg=zeros(1,co);
diffb=zeros(1,co);
diffh=zeros(1,co);
for i=1:co
    num=stats(i).FilledImage;
    [r c]=size(num);
    s=sum(sum(num));
    mm_green(1,i)=i_green(1,i)/s;

    temp2=stats(i).SubarrayIdx;
    temp3=temp2{1,1};
    temp4=temp2{1,2};
    ceny=round(r/2);
    cenx=round(c/2);
    r1=temp3(1,ceny)-6;
    rm=temp3(1,ceny)+6;
    c1=temp4(1,cenx)-6;
    cm=temp4(1,cenx)+6;
    if c1<=0
        c1=1;
    end
    if r1<=0
        r1=1;
    end
    rrr=0;
    gg=0;
    bb=0;
    hh=0;
    if rm>R
        rm=R;
    end
    if cm>C
        cm=C;
    end
end

```

```

for p=r1:rm
    for q=c1:cm
        rrr=rrr+Redinput(p,q);
        gg=gg+Greeninput(p,q);
        bb=bb+Blueinput(p,q);
        hh=hh+Hueinput(p,q);
    end
end
meanr=rrr/(r1*c1);
meang=gg/(r1*c1);
meanb=bb/(r1*c1);
meanh=hh/(r1*c1);
diffrr(1,i)=meanr-mm_green(1,i);
diffgg(1,i)=meang-mm_green(1,i);
diffbb(1,i)=meanb-mm_green(1,i);
diffhh(1,i)=meanh-mm_green(1,i);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Feature 30 (MajorAxis)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MajorAxis = [stats.MajorAxisLength];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Feature 31 (MinorAxis)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MinorAxis = [stats.MinorAxisLength];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save('FeatureFile.mat');

```

تابع MaskGeneration

```

function [mask]=MaskGeneration(image,thresh)
%-----
% Programmer: Elaheh Imani 1392/12/19
%-----
mask=image(:,:,2)>thresh;
se=strel('disk',5);
mask=imopen(mask,se);
mask=imclose(mask,se);
clear 'se';
end

```

تابع locradfun1

```

function [mx LStrt LEnd LineAngle LineVld]=locradfun1(f)
global BlkMsk TetaStp;
%figure(1); imshow(f(:,:,2));
Level=0.7; %#####Line Width Control

[R C D]=size(f);
f=double(f);
g1=255-f(:,:,2);
g1=g1/255; g1=imadjust(g1);
immean = mean2(g1);
g1=g1.*BlkMsk;
%figure(9); imshow(g1);

theta = 0:TetaStp:180;
[Rd,xp] = radon(g1,theta);

```



```

h=fspecial('average',3);
Rd=imfilter(Rd,h);

[mv mi]=max(Rd);
[mx mc]=max(mv);
mr=mi(mc);

LineAngle=mc*TetaStp;

if LineAngle > 90
    OrtAngle=fix((LineAngle-90)/TetaStp);
else
    OrtAngle=fix((LineAngle+90)/TetaStp);
end

maxcul=Rd(:,mc);
maxcul=maxcul-R*immean;    maxcul=maxcul.*((sign(maxcul)+1)/2);
%figure(2); plot(maxcul);
[mx LinePos]=max(maxcul);

if mx == 0
    LineVld=0;
    LStrt=0;
    LEnd=0;
else
    %LinePos=mr;
    LineVld=mx/R;
    [RR RC]=size(Rd);

    LStrt=LinePos;
    LEnd=LinePos;
    for i=LinePos:-1:1
        LStrt=i;
        if maxcul(i)<Level*mx
            break
        end
    end
    for i=LinePos:1:RC
        LEnd=i;
        if maxcul(i)<Level*mx
            break
        end
    end

    LStrt=LStrt-fix(RR/2)+fix(R/2);
    LEnd=LEnd-fix(RR/2)+fix(R/2);
    %[LStrt LEnd]
    if LStrt<1
        LStrt=1;
    end
    if LEnd>R
        LEnd=R;
    end
    %[LStrt LEnd]
end

```

تابع CorrAB

```
function rA = CorrAB(A,B)
%-----
% Programmer: H.Mansoori
%-----

[R,C] = size(A);
[R2,C2] = size(B);
Rw2 = (R2-1)/2;
Cw2 = (C2-1)/2;
Ro = Rw2+1;
Co = Cw2+1;
meanA = zeros(R-2*Rw2,C-2*Cw2);
for i=-Rw2:Rw2
    for j=-Cw2:Cw2
        meanA = meanA + A(1+Rw2+i:R-Rw2+i,1+Cw2+j:C-Cw2+j);
    end
end
meanA = meanA/(R2*C2);
Bhat = B - mean2(B);
SA = zeros(R-2*Rw2,C-2*Cw2);
MA = zeros(R-2*Rw2,C-2*Cw2);
for i=-Rw2:Rw2
    for j=-Cw2:Cw2
        SA = SA + (A(1+Rw2+i:R-Rw2+i,1+Cw2+j:C-Cw2+j)-
meanA).*(Bhat(Ro+i,Co+j));
        MA = MA + (A(1+Rw2+i:R-Rw2+i,1+Cw2+j:C-Cw2+j)-meanA).^2;
    end
end
MA = (MA.*(sum(sum(Bhat.^2))))).^0.5;
rA = SA./MA;

end
```

تابع ColorNormalization

```
function [img]=ColorNormalization(image,mask>window)
%-----
% color normalization of retinal images
%-----
% Input:
%   window: window size
%
% Output:
%   img: normalized retinal image
%-----
% Programmer: Elaheh Imani 1393/2/10
%-----
%   global image;
%   global mask;
b=load('color_normalization');
Mu=b.Mu;
STD=b.STD;
clear b
```

```

min_value=0;max_value=255;
R=image(:,:,1);
G=image(:,:,2);
B=image(:,:,3);
window_size=round(size(image,1)/window);
R_filtered=double(medfilt2(R,[window_size window_size]));
G_filtered=double(medfilt2(G,[window_size window_size]));
B_filtered=double(medfilt2(B,[window_size window_size]));

R_filtered =(double(R)-R_filtered);
G_filtered = (double(G)-G_filtered);
B_filtered = (double(B)-B_filtered);
clear R G B
m_r=min(R_filtered(mask(:)));
m_g=min(G_filtered(mask(:)));
m_b=min(B_filtered(mask(:)));

R_filtered=min_value+((R_filtered-m_r)./(max(R_filtered(mask(:)))-
m_r))*max_value;
G_filtered=min_value+((G_filtered-m_g)./(max(G_filtered(mask(:)))-
m_g))*max_value;
B_filtered=min_value+((B_filtered-m_b)./(max(B_filtered(mask(:)))-
m_b))*max_value;
clear m_r m_g m_b min_value max_value
Mu_r=mean(R_filtered(mask(:)));
Mu_g=mean(G_filtered(mask(:)));
Mu_b=mean(B_filtered(mask(:)));

STD_r=std(R_filtered(mask(:)));
STD_g=std(G_filtered(mask(:)));
STD_b=std(B_filtered(mask(:)));

R_filtered=(R_filtered-Mu_r)./STD_r;
G_filtered=(G_filtered-Mu_g)./STD_g;
B_filtered=(B_filtered-Mu_b)./STD_b;
clear STD_r STD_g STD_b Mu_r Mu_g Mu_b
new_r=(R_filtered*STD(1))+Mu(1);
new_g=(G_filtered*STD(2))+Mu(2);
new_b=(B_filtered*STD(3))+Mu(3);
clear R_filtered G_filtered B_filtered STD Mu
new_r(~mask)=0;
new_g(~mask)=0;
new_b(~mask)=0;
img=zeros(size(image),'uint8');
img(:,:,1)=new_r;
img(:,:,2)=new_g;
img(:,:,3)=new_b;
clear new_r new_g new_b
end

```

تابع xml2struct

```

function [ s ] = xml2struct( file )
%Convert xml file into a MATLAB structure
% Modified by X. Mo, University of Wisconsin, 12-5-2012

```

```

    if (nargin < 1)
        clc;
        help xml2struct
        return
    end

    if isa(file, 'org.apache.xerces.dom.DeferredDocumentImpl') || isa(file,
'org.apache.xerces.dom.DeferredElementImpl')
        % input is a java xml object
        xDoc = file;
    else
        %check for existence
        if (exist(file,'file') == 0)
            %Perhaps the xml extension was omitted from the file name. Add
the
            %extension and try again.
            if (isempty(strfind(file, '.xml')))
                file = [file '.xml'];
            end

            if (exist(file,'file') == 0)
                error(['The file ' file ' could not be found']);
            end
        end
        %read the xml file
        xDoc = xmlread(file);
    end

    %parse xDoc into a MATLAB structure
    s = parseChildNodes(xDoc);

end

% ----- Subfunction parseChildNodes -----
function [children,ptext,textflag] = parseChildNodes(theNode)
    % Recurse over node children.
    children = struct;
    ptext = struct; textflag = 'Text';
    if hasChildNodes(theNode)
        childNodes = getChildNodes(theNode);
        numChildNodes = getLength(childNodes);

        for count = 1:numChildNodes
            theChild = item(childNodes,count-1);
            [text,name,attr,childs,textflag] = getNodeData(theChild);

            if (~strcmp(name,'#text') && ~strcmp(name,'#comment') &&
~strcmp(name,'#cdata_dash_section'))
                %XML allows the same elements to be defined multiple times,
                %put each in a different cell
                if (isfield(children,name))
                    if (~iscell(children.(name)))
                        %put existings element into cell format
                        children.(name) = {children.(name)};
                    end
                end
            end
        end
    end
end

```

```

        end
        index = length(children.(name))+1;
        %add new element
        children.(name){index} = childs;
        if(~isempty(fieldnames(text)))
            children.(name){index} = text;
        end
        if(~isempty(attr))
            children.(name){index}.('Attributes') = attr;
        end
    else
        %add previously unknown (new) element to the structure
        children.(name) = childs;
        if(~isempty(text) && ~isempty(fieldnames(text)))
            children.(name) = text;
        end
        if(~isempty(attr))
            children.(name).('Attributes') = attr;
        end
    end
end
else
    ptextflag = 'Text';
    if (strcmp(name, '#cdata_dash_section'))
        ptextflag = 'CDATA';
    elseif (strcmp(name, '#comment'))
        ptextflag = 'Comment';
    end

    %this is the text in an element (i.e., the parentNode)
    if (~isempty(regexprep(text.(textflag), '[\s]*', '')))
        if (~isfield(ptext, ptextflag) ||
            isempty(ptext.(ptextflag)))
            ptext.(ptextflag) = text.(textflag);
        else
            %what to do when element data is as follows:
            %<element>Text <!--Comment--> More text</element>

            %put the text in different cells:
            % if (~iscell(ptext)) ptext = {ptext}; end
            % ptext{length(ptext)+1} = text;

            %just append the text
            ptext.(ptextflag) = [ptext.(ptextflag)
text.(textflag)];
        end
    end
end
end
end
end

% ----- Subfunction getNodeData -----
function [text,name,attr,childs,textflag] = getNodeData(theNode)
    % Create structure of node info.

```

```

%make sure name is allowed as structure name
name = toCharArray(getNodeName(theNode));
name = strrep(name, '-', '_dash_');
name = strrep(name, ':', '_colon_');
name = strrep(name, '.', '_dot_');

attr = parseAttributes(theNode);
if (isempty(fieldnames(attr)))
    attr = [];
end

%parse child nodes
[childs,text,textflag] = parseChildNodes(theNode);

if (isempty(fieldnames(childs)) && isempty(fieldnames(text)))
    %get the data of any childless nodes
    % faster than if any(strcmp(methods(theNode), 'getData'))
    % no need to try-catch (?)
    % faster than text = char(getData(theNode));
    text.(textflag) = toCharArray(getTextContent(theNode));
end

end

% ----- Subfunction parseAttributes -----
function attributes = parseAttributes(theNode)
    % Create attributes structure.

    attributes = struct;
    if hasAttributes(theNode)
        theAttributes = getAttributes(theNode);
        numAttributes = getLength(theAttributes);

        for count = 1:numAttributes
            %attrib = item(theAttributes,count-1);
            %attr_name = regexprep(char(getName(attrib)),'[-:.]','_');
            %attributes.(attr_name) = char(getValue(attrib));

            %Suggestion of Adrian Wanner
            str = toCharArray(toString(item(theAttributes,count-1)));
            k = strfind(str,'=');
            attr_name = str(1:(k(1)-1));
            attr_name = strrep(attr_name, '-', '_dash_');
            attr_name = strrep(attr_name, ':', '_colon_');
            attr_name = strrep(attr_name, '.', '_dot_');
            attributes.(attr_name) = str((k(1)+2):(end-1));
        end
    end
end
end

```

- [1] Zhang, Bob, et al. "Detection of microaneurysms using multi-scale correlation coefficients." *Pattern Recognition* 43.6 (2010): 2237-2248.
- [2] Gardner, G. G., Keating, D., Williamson, T. H. and Elliott, A. T., "Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool," *Brit. J. Ophthalmol.* 80(11), 940–944 (1996).
- [3] Estabridis, K. and de Figueiredo, R. J. P., "Automatic Detection and Diagnosis of Diabetic Retinopathy," *Proc. IEEE International Conference on Image Processing*, 445–448 (2007).
- [4] Niemeijer, M., van Ginneken, B., Staal, J., Suttorp-Schulten, M. S. A. and Abramoff, M. D., "Automatic detection of red lesions in digital color fundus photographs," *IEEE Trans. Medical Imaging* 24(5), 584–592 (2005).
- [5] Spencer, T., Olson, J. A., McHardy, K. C., Sharp, P. F. and Forrester, J. V., "An image processing strategy for the segmentation of microaneurysms in fluorescein angiograms of the ocular fundus," *Computers and Biomedical Research* 29(4), 284-302 (1996).
- [6] Niemeijer, M., van Ginneken, B., Staal, J., Suttorp-Schulten, M. S. A. and Abramoff, M. D., "Automatic detection of red lesions in digital color fundus photographs," *IEEE Trans. Medical Imaging* 24(5), 584–592 (2005).
- [7] M. Kalaivani¹, M. S. Jeyalakshmi², Aparna.V³; Extraction Of Retinal Blood Vessels Using Curvelet Transform And Kirsch's Templates ; *International Journal of Emerging Technology and Advanced Engineering* Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 11, November 2012)