

# دکتر: "کمک"

نویسنده: مهدی رئیسی  
عضو انجمن دانش‌آموزی پایتون  
خانه ریاضیات اصفهان

در این خواندنی می‌خوایم یکم به دکترها برای تشخیص سرطان پوستی که اسمش ملانوما هست کمک کنیم. حالا یه سوال، دکترها چطوری تشخیص میدن؟ مگه تشخیص دکتر چه مشکلی داره که از کامپیوتر بخوایم تشخیص بده یا کمکمون کنه که تشخیص بدیم؟ 😊

اول بیاین به مشکل بزرگ که از گذشته‌های دور بوده اشاره کنیم ( ترس از دکتر 😞 ) چرا از دکتر می‌ترسیم؟ چون دکترها درد دارن. در واقع آزمایش‌ها و عمل‌ها و .... درد دارن و یه مشکل بزرگ‌تر اینه که اگه سالم هم باشی باید درد آزمایش‌ها رو تحمل کنی.

ولی عکس گرفتن درد نداره (در واقع هیچی حس نمی‌کنید 😊) ( بیایید یکی از اون آزمایش‌ها رو به راحتی عکس گرفتن کنیم.

حالا که فهمیدیم کار ما کمک بزرگی به بشریت می‌کنه، ببینیم این بیماری اصلا چی می‌گه! سرطانی که ما می‌خوایم بررسی کنیم، با چشم هم قابل دیدنه 😊 و شبیه یه لکه روی پوست هست (ولی هر لکه ای غده سرطانی نیست)

ما می‌خوایم به دکتر بگیم از اون لکه‌ای که روی پوست مریض هست، یه عکس بگیرن به برنامه‌ی ما بدن تا ما کمک کنیم تا بفهمن آیا این یک غده‌ی سرطانی هست 😞 یا نه. 😊

کمک ما چی هست؟ و چه جوابی به عنوان جواب آزمایش به دکتر می‌خوایم بدیم؟؟

اول لکه رو توی عکس پیدا می‌کنیم و محیط و مساحت لکه رو به واحد پیکسل محاسبه می‌کنیم. بعد یک دایره می‌کشیم که مساحتش با مساحت لکه یکی باشه. بعد نسبت محیط لکه به محیط دایره‌ی هم مساحتش رو محاسبه می‌کنیم و اون مقدار رو **ratio** نامگذاری می‌کنیم.

حالا هم مساحت و هم محیط و هم **ratio** رو به دکتر اعلام می‌کنیم.

به همین سادگی 😊

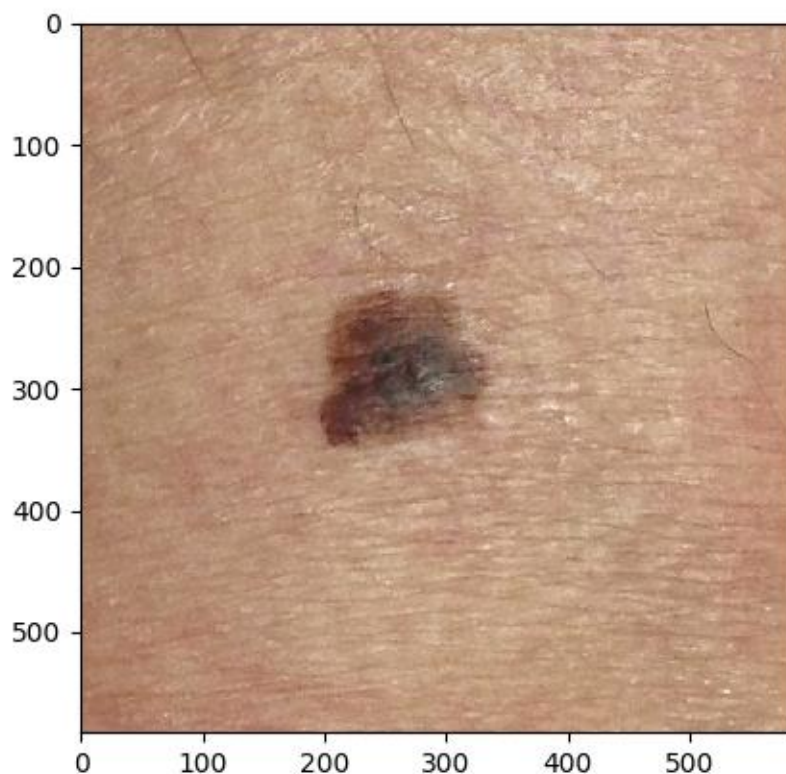
ولی حالا دکتر از این مقدار چی می‌فهمه؟؟

اگر این مقدار زیاد باشه معنیش چیه؟؟ یعنی محیط لکه نسبت به مساحتش خیلی زیاده و این یعنی تورفتگی‌ها و ناهمواری‌های زیادی روی محیط لکه وجود داره 😊

حالا باید در مورد داده‌ها که همان عکس‌ها هستند صحبت کنیم.

ما تعداد زیادی عکس از لکه‌های پوستی داریم که دکترها با روش‌های خیلی دردناک، اونها رو بررسی کردن و به ما گفتن که آیا سرطانی هستن یا امکانش وجود داره که سرطانی باشن و یا امکانش خیلی کمه که سرطانی باشن.

در این خواندنی قصد داریم عکس زیر را بررسی کنیم.



حالا یکم درمورد عکس توضیح بدیم (منظورم از عکس هر عکسیه، حتی عکس شب تولد شما دوست عزیز) هر عکس از نقاط خیلی خیلی کوچکی تشکیل شده که بهش پیکسل گفته میشه. هر کدوم از این نقاط فقط یک رنگ می تونه داشته باشه مثلاً فقط آبی.

برای اینکه بهتر بفهمین اگر توی خونه هستین زیر پاتون رو نگاه کنید، چی می بینید؟ (قالی؟) اگه کمی ریز بشین می بینید که یک عالمه نخ هستن که ما فقط سرشون رو می بینیم و هر نخ یه رنگ هست و وقتی از دور به قالی نگاه می کنیم یه تصویر رنگارنگ می بینیم. اون نخ ها همون پیکسل های قالی هستن. عکس هم دقیقاً از چنین چیزایی تشکیل شده.

حالا که با پیکسل آشنا شدیم بریم سراغ رنگ ها!

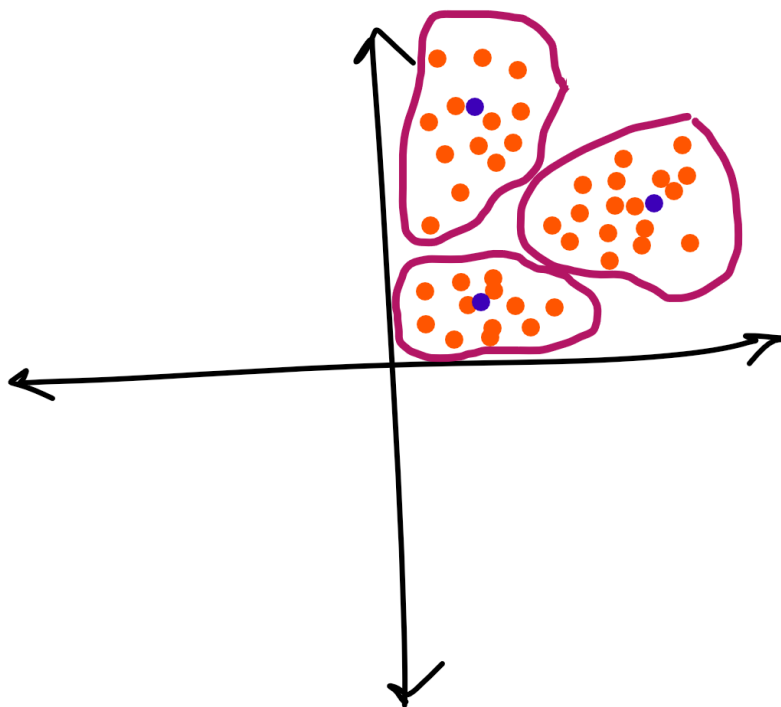
هر رنگ از مجموع سه رنگ آبی، قرمز و سبز تشکیل شده (RGB). در واقع هر رنگ رو میشه به این سه رنگ تجزیه کرد. پیکسل، سه مقدار داره که هر مقدار نشان دهنده ی مقدار هر یک از سه رنگ اصلی هست. هر مقدار یک عدد صحیح از ۰ تا ۲۵۵ هست (یعنی ۲۵۶ رنگ) و چون سه تا از این مقدارها در هر پیکسل وجود داره  $۲۵۶ * ۲۵۶ * ۲۵۶$  حالت (رنگ) مختلف برای هر پیکسل وجود داره. یعنی چند تا؟؟ یعنی خیلی زیاد و این خودش کار پردازش رو خیلی سخت می کنه و تقریباً امکان نداره در یک عکس از شب تولد شما همه ی بخش های لباس شما در عکس یک رنگ داشته باشن. یعنی پیکسل هایی که دارن عکس شما رو نمایش میدن همه یک رنگ داشته باشن (هر چند که کل لباس شما یک رنگ داره) خوب اگه گفتین چرا اینطوری میشه؟؟

برای همین اولین کاری که ما می خوایم بکنیم، این هست که دامنه ی (تعداد حالات ممکن) رنگ های موجود در عکس رو کاهش بدیم. ولی به چند رنگ؟؟ با توجه به اینکه عکس ها فقط بخشی از پوست و کل لکه رو نشون میدن به ۳ رنگ کاهش میدیم.

چرا ۳ تا؟؟ خوب یکی رنگ پوست یکی رنگ لکه و یکی هر چیز دیگری مثل نویز در عکس.

برای این کار من از الگوریتم KMeans استفاده کردم.

روشی برای تقسیم کردن. فرض کنید یک عالمه نقطه در یک صفحه مختصات دارین و می خواین این نقطه ها رو به ۳ دسته تقسیم کنید. یعنی نقاطی که نزدیک به هم هستن رو در یک دسته قرار بدیم و یک سرگروه که وسط همه ی اونها هست رو انتخاب کنیم. از این به بعد اگه پرسیدن فلان نقطه مختصاتش چیه؟ مختصات سرگروه رو اعلام می کنیم.



به نمودار بالا دقت کنید، آبی‌ها سرگروه‌ها، دسته‌ها هم مشخص شده. حالا اگر مختصات هر کدوم از اون نقاط رو بپرسن، مختصات سرگروه رو می‌گیریم. در واقع ما سه جواب بیشتر برای گفتن نداریم. (۴۶)

حالا برای رنگ‌ها هم همین کار رو می‌کنیم. هر پیکسل رو یک نقطه در نظر می‌گیریم و هر کدوم از اون سه رنگ رو یکی از بُعد هاش قرار می‌دیم. یعنی رنگ رو در یک صفحه‌ی مختصات سه بعدی نمایش می‌دیم (۳ بعدی شد). اشکالی نداره هم  $X$  داره هم  $Y$  داره هم  $Z$  که همان مقدار  $R, G, B$  پیکسل ما هستند. خوب اینها رو خوشه‌بندی می‌کنیم. (همون گروه گروه کردن) و به جای رنگ هر کدوم از پیکسل‌ها، رنگ سرگروه رو قرار می‌دیم. یعنی عکس ما فقط سه نوع رنگ خواهد داشت.

اینکه الگوریتم چطوری این کار رو انجام میده رو شما باید برین تحقیق کنید. (خیلی سخت نیست کافیه یه سرچ انگلیسی کوچیک کنید)

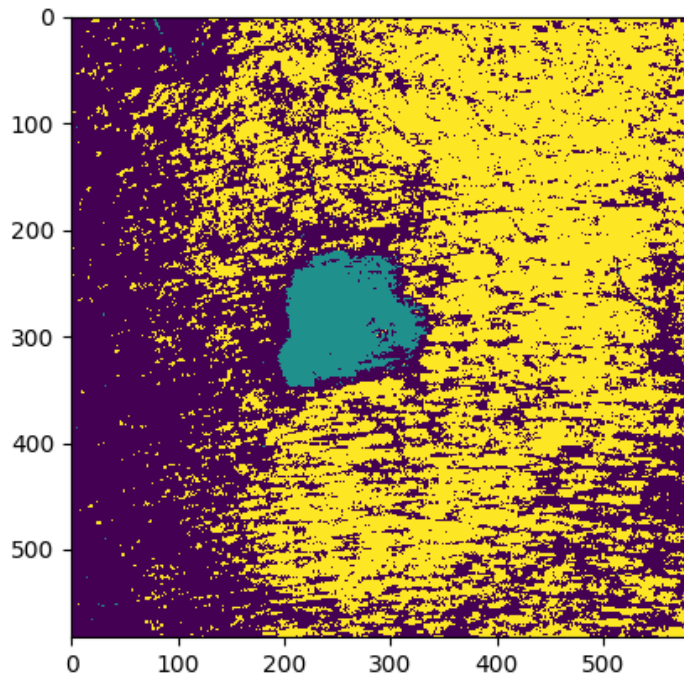
من در یک فایل پایتون برنامه‌ای نوشتم که این کار رو انجام میده. اسم فایل `Clustering` هست.

در این فایل یه کلاس هست که دوتا ورودی داره. یکی `image` که همون عکسه و یکی هم `n_clusters` که تعداد خوشه‌هاست و به‌طور پیش‌فرض مقدارش ۳ هست.

متد `run` رو اجرا کنید. اولین خروجی این متد عکس و دومی مرکز خوشه‌هاست.

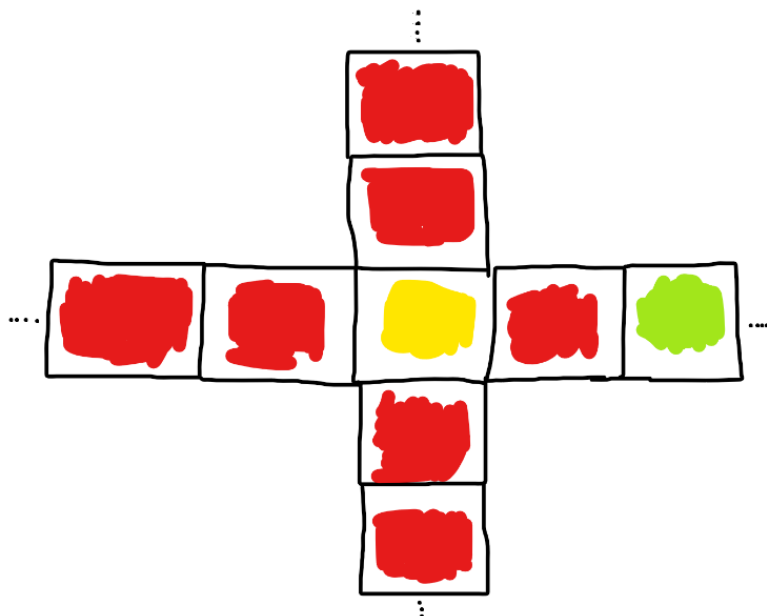
برای مشاهده عکس تبدیل شده، متد `show_image` رو از این کلاس اجرا کنید.

حالا خروجی رو برای عکس ما ببینید.



برای نمایش از ۳ رنگ فرضی استفاده شده (برای اینکه بهتر نمایش بدیم) ولی در واقعیت هر پیکسل به رنگ سرگروهش هست. کار بعدی چیه؟؟

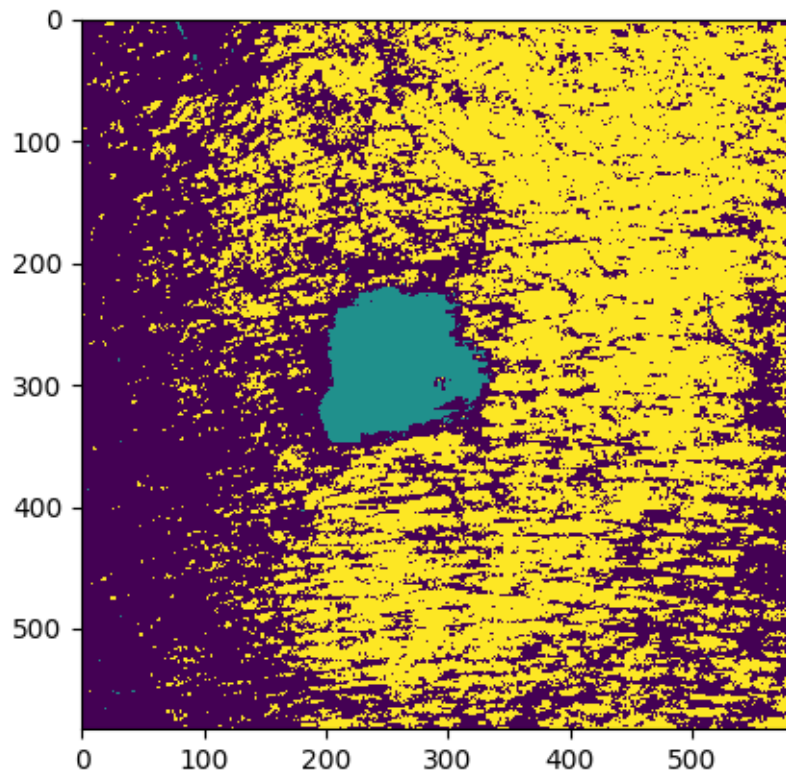
اوه خدای من، یک جاهایی وسط لکه عضو لکه نیست و سرگروهش رنگ پوست هست. حالا چه کار کنیم؟؟  
 برای هر پیکسل همسایه‌هایش رو بررسی می‌کنیم که عضو کدام گروه هستن. اگر ۷۰ درصد عضو یک گروه دیگه بودن، پیکسل هم عضو اون گروه میشه.



عکس بالا رو نگاه کنید. ناحیه زرد رو می‌خوایم بررسی کنیم ( زرد رنگ سر گروه‌هشه) همسایه هاش رو نگاه می‌کنیم می‌بینیم بیشترشون قرمز هستن. به این نتیجه می‌رسیم که این یکی هم قرمز هست. (ولی حالا چرا تا ۲ تا همسایه اون طرفتر؟؟ نمیشد در شکل بیشتر از این رو جا بدم ولی در برنامه تا ۴ تا اون طرفتر رو بررسی کردم. این طوری شد که کلی از پیکسل‌های فراری از گروه رو به گروه خودشون برگردوندیم و به یک لکه تقریبا خوب رسیدیم.

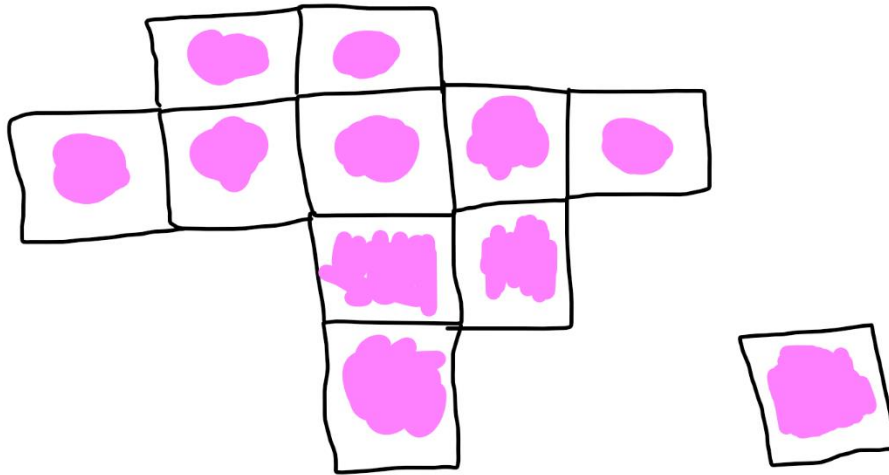
اسم فایل پایتونی که این کارا رو انجام میده `smooth` هست. که یک کلاس به همین اسم داره و ورودی اولش عکسی هست که خروجی فایل پایتون قبلی هست. دومی هم سرگروه‌ها هستند. سومی هم تعداد همسایه‌هایی که می‌خوایم بررسی بشه ( همون  $m$ ). و آخری درصد تفاوت همسایه‌ها هست. اسم ورودی  $n$  هست مثلا ۷۰ در صد رو باید بهش `0.7` بدیم.

اینجا هم مثل کلاس قبلی یه متد `run` و یه متد `show_image` داریم که خروجی `run` فقط عکس جدید هست.



سوال: ۳ تا گروه داریم کدوم گروه لکه هست؟؟ اون که از همه تیره‌تره. کدوم از همه تیره‌تره؟؟ 😊 مقدار `RGB` رنگ سیاه `[0,0,0]` هست یعنی هیچ رنگی وجود نداشته باشه پس هر گروهی که سرگروهش به این مقادارها نزدیک‌تر باشه اون گروه تیره‌تره (میانگین مربعات به هم نزدیک باشه) گروهی که تیره‌تر باشه گروه لکه هست 😊

خوب لکه رو پیدا کردیم. ولی در فیلتری که هر فراری رو به گروه خودش برگردوندیم همه‌ی فراری‌ها به خوشه (گروه) خودشون برنگشتن و هنوز هم فراری‌هایی وجود داره که دیگه باید بیخیالشون بشیم. عکس بالا رو نگاه کنید. سمت چپ بالا، چند تا پیکسل هستند که عضو گروه لکه هستن ولی هیچ ربطی به لکه ندارن و با فاصله‌ی زیادی از اون قرار گرفتن 😞 همین طوری رهاشون نمی‌کنیم. در خوشه‌ی لکه اونهایی که به بقیه چسبیده نیستن رو حذف می‌کنیم.

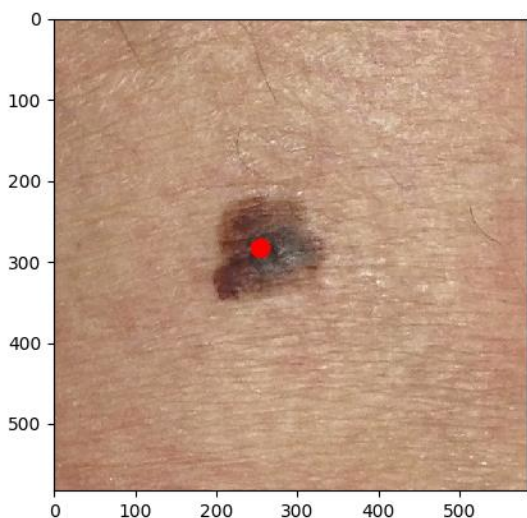


مثلا در عکس بالا مرکز خوشه (سرگروه) برای همه بنفش هست. ولی مثل اینکه یکی از اینها به بقیه چسبیده نیست پس ما اون رو از خوشه حذف می‌کنیم.

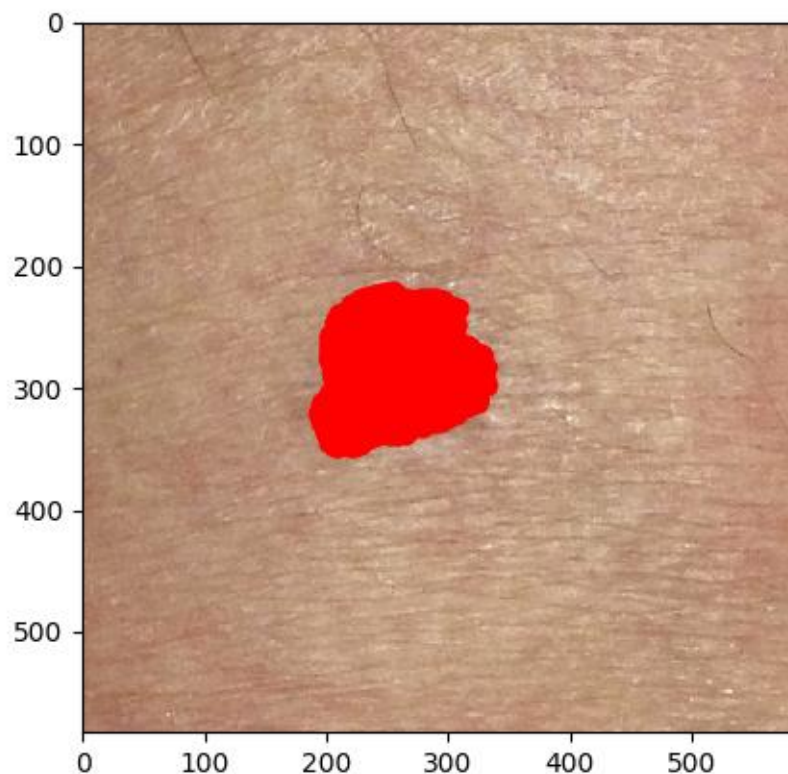
یه سوال!

آیا امکان داره دو دسته از پیکسل‌ها هر کدوم از چند تا پیکسل به هم چسبیده درست شده باشن؟ برای اینکه بفهمیم کدوم یکی لکه هست و کدوم لکه نیست میایم و مرکز تجمع رو پیدا می‌کنیم. یعنی جایی که بیشتر پیکسل‌های لکه جمع شدن (مثل جاهایی که یه چیزی رو رایگان میدن و یهویی شلوغ میشه و تجمع میشه)

من از طریق یه الگوریتم نه چندان سخت اون نقطه تجمع رو پیدا کردم. البته منظورم میانگین نیست 😊 ولی از میانگین در الگوریتم استفاده کردم) اگه دوست داشتن می‌تونید برین و برنامه رو یه بار بخونید.



حالا از اون مرکز شروع می‌کنیم و تمام همسایه‌هایی که بهش چسبیدن رو پیدا می‌کنیم.



خوب ما لکه رو پیدا کردیم و می‌تونیم مساحتش رو حساب کنیم.

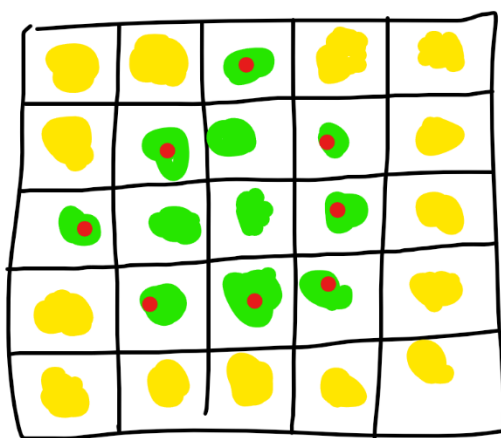
من این دو کار رو در دو فایل انجام دادم. یکی `find_median` و یکی هم `find_mole` که از اسم هر کدوم پیدا است چه کاری انجام میدن. مثل کلاس‌های قبلی هم `run` دارن و هم `show_image` و این یک قانونه (البته در این برنامه)

پس محیط چی 

بریم محیط رو محاسبه کنیم. حالا کدوم پیکسل‌ها پیکسل‌های محیط هستن؟؟

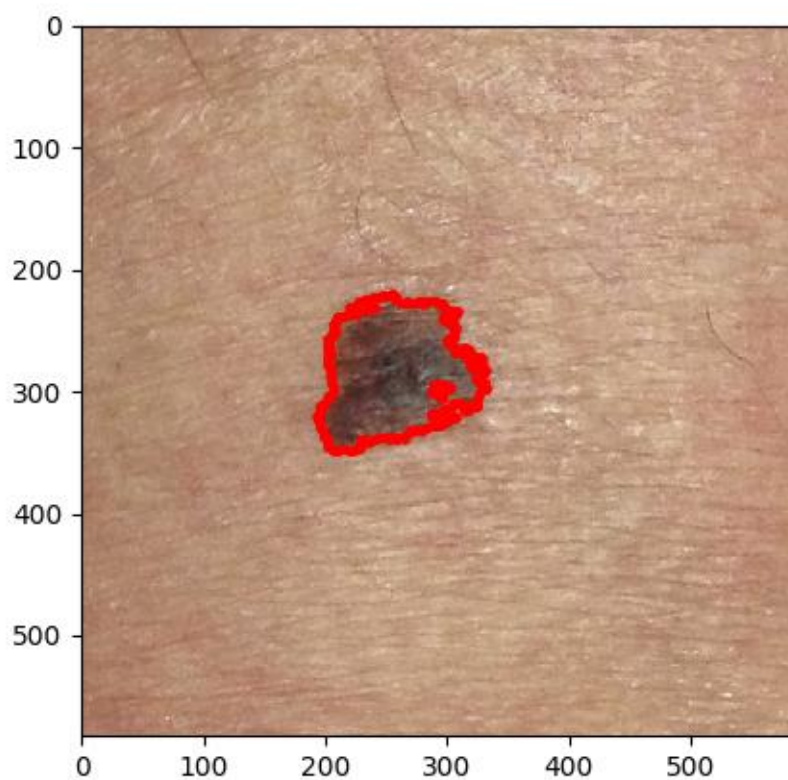
اون پیکسل‌هایی از لکه که برچسب یکی از همسایه‌های دیوار به دیوارش (سرگروه) با برچسب لکه تفاوت داشته باشه .





مثلا در عكس بالا سبزه‌ها لکه هستن و اونهایی که نقطه قرمز دارن محیط هستن.

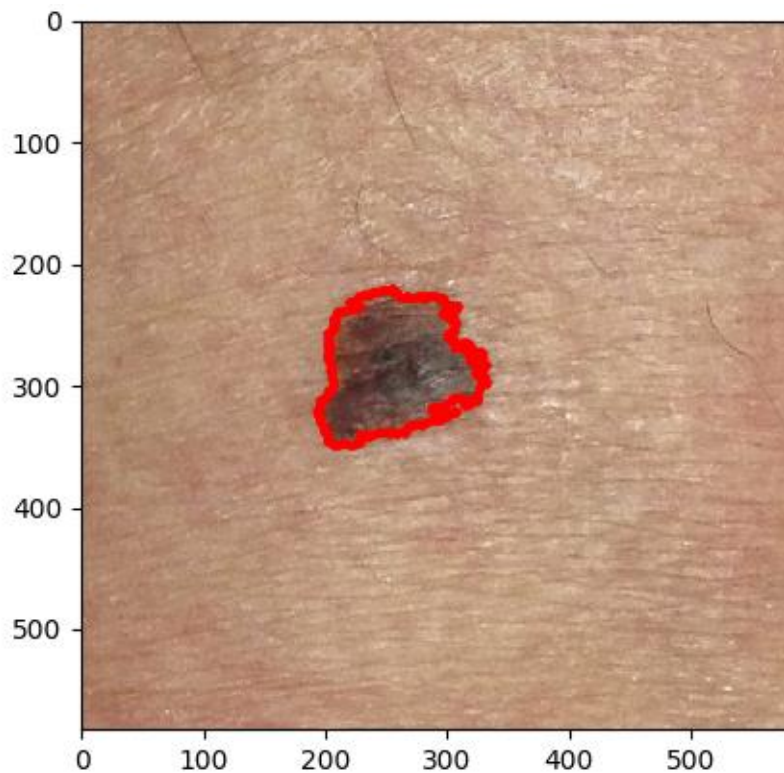
همین کار رو برای عکسمون انجام میدیم.



چی شد !!! یکی از اون نویزها باعث شد بخش‌هایی که محیط نیستند رو هم محیط حساب کنیم.

باید دنبال خط‌های دنباله‌دار بگردیم. مثل چی؟؟ فرض کنید یک دسته آدم داریم که بعضی‌ها در صف ایستادن و بعضی با کمی فاصله از اون صف ایستادن، یا حتی یک صف کوچک دیگه تشکیل دادن تا هر موقع که شد بپرن وسط صف اصلی. می‌خوایم صف اصلی رو پیدا کنیم. چه کار کنیم؟ دنبال خط می‌گردیم و اگه دوتا خط دنباله دار شد اونوی رو دنبال می‌کنیم که بزرگتره. و با این روش محیط واقعی را پیدا می‌کنیم ( البته اینجا الگوریتم من برای بعضی عکس‌ها خوب جواب نمیده و همون فیلتر نشدش بهتر کار می‌ده)

در فایل `perimeter` محیط رو پیدا کردم و در فایل `filter_perimeter` به دنبال خط‌های دنباله‌دار گشتم.



حالا به دکتر نتیجه رو اعلام می‌کنیم.

و برای محاسبه از فایل `compute_ratio` استفاده می‌کنیم.

شاید بخواین یکم بیشتر درمورد برنامه نویسی این برنامه بدونید.

در این برنامه که به زبان پایتون نوشته شده چند کتابخانه استفاده شده که در پایتون نیست و باید اونها رو نصب کنی مثل:

Numpy  
Scipy  
Matplotlib  
SKLearn

برای اجرای برنامه به این کتابخانه‌ها نیاز هست.

## برنامه کجاست ؟

برنامه در **github** هست که می‌تونید به راحتی اون رو از لینک زیر دانلود کنید.

<https://github.com/MMReisi/Melanoma>

پیشنهاد می‌کنم از **Fork** استفاده کنید. چون احتمالاً در برنامه تغییرات خوبی ایجاد خواهد شد.

پروژه رو که دانلود کردید، اسمش **Melanoma** هست. پوشه رو باز کنید، یک پوشه با نام **Samples\_images** داخلش هست که حاوی کلی عکس هست. ( عکسهایی که دکتر به ما داده )

یک پوشه دیگه با نام **question** هست که داخلش فایل‌هایی هست که با کمک اون برنامه رو نوشتیم. این فایل متعلق به دانشگاه پلی‌تکنیک ایتالیاست.

در پوشه یک فایل دیگه هم گذاشتم که اسمش **runner.py** هست. اجراش کنید و اسم عکس رو وارد کنید (از همون عکس‌هایی که در پوشه **samples\_images** هست) فقط فراموش نکنید که فرمت رو هم بنویسید (مثلاً **low\_risk\_1.jpg**). برنامه یکم کند هست. پس کمی صبور باشید تا خروجی برنامه نمایش داده بشه. عکس‌ها رو به ترتیب نشون میده. برای رفتن از هر عکس به عکس بعدی باید عکسی که باز هست رو ببندید.

امیدوارم خوشتون اومده باشه ☺

