

## باسمه تعالی

### پروژه تشخیص چهره

برنامه‌ای بنویسید که نام یک فایل تصویری با پسوند BMP را از کاربر بپرسد و تعیین کند که آیا در این تصویر، چهره‌ای وجود دارد یا خیر. در صورت مثبت بودن جواب، مکان تقریبی چهره‌ها را تعیین کنید.

برای این منظور باید از الگوریتم‌های تشخیص چهره استفاده کنید. یک الگوریتم ساده برای این کار در ادامه توضیح داده می‌شود. البته شما می‌توانید از هر الگوریتم دیگری نیز استفاده کنید. اما برای سادگی، استفاده از این الگوریتم پیشنهاد می‌شود. کافی است مراحل توضیح داده شده را قدم به قدم انجام دهید تا شما هم بتوانید یک برنامه تشخیص چهره داشته باشید!

برای دریافت اسلایدهای توضیح پروژه، [اینجا](#) را کلیک کنید.

توضیح مقدماتی:

هر فایل BMP، یک فایل باینری محسوب می‌شود که اطلاعات مربوط به تصویر، به صورت تعدادی عدد صحیح در آن قرار دارد. کافی است بدانید که هر عدد در فایل باینری BMP به چه معناست. الگوی فایل‌های BMP را می‌توانید به راحتی در اینترنت جستجو کنید.

در دنیای کامپیوتر، برای بیان کردن رنگ‌ها از چند سیستم مختلف استفاده می‌شود. معروفترین آن‌ها سیستم RGB است. در این سیستم، رنگ هر پیکسل با سه عدد یک بایتی (سه عدد از ۰ تا ۲۵۵) مشخص می‌شود. این سه عدد به ترتیب بیان‌کننده میزان رنگ قرمز، سبز و آبی در تشکیل رنگ نهایی است. مثلاً سه‌تایی (150,0,130) نشانگر رنگی است که در آن اثری از سبز وجود ندارد و ترکیبی از قرمز و آبی با نسبت مشخص است که احتمالاً چیزی شبیه به بنفش خواهد بود. دو سیستم دیگر بیان رنگ‌ها، یکی نرمالایز شده RGB است (یا همان rg) و دیگری HSL. فرمول تبدیل RGB به rg در اسلاید آمده است. برای تبدیل RGB به HSL نیز برای سهولت کار شما، تابعی نوشته شده است

که می‌توانید آن را از [اینجا](#) دریافت و در برنامه‌تان استفاده کنید. این کد به زبان MATLAB است. باید آن را به زبان C تبدیل کنید.

الگوریتم تشخیص چهره بدین ترتیب است که شما باید تمام پیکسل‌های موجود در تصویر را بررسی کنید و پیکسل‌هایی را که می‌توانند تصویر چهره باشند را مسخ‌کنید. این کار در یک فرآیند چند مرحله‌ای انجام می‌شود و هر بار بر اساس برخی معیارها، شما تعدادی از پیکسل‌های زائد را کنار می‌گذارید (سیاه می‌کنید) تا در نهایت فقط پیکسل‌هایی باقی‌مانند که نشانگر یک چهره هستند.

مراحل انجام پروژه:

۱- نام فایل BMP را از کاربر بپرسید.

۲- اطلاعات مربوطه را از فایل استخراج کنید تا رنگ تمام پیکسل‌ها معلوم شود.

۳- رنگ هر پیکسل را به سیستم HSL و rg تبدیل کنید.

۴- هر پیکسل را بر اساس فرمول‌های مرحله اول موجود در اسلاید بررسی کنید (محدوده پارامتر H، مقدار F1، F2 و W).

۵- یک تصویر سیاه و سفید رسم کنید که در آن پیکسل‌های نامرتب‌همگی به رنگ سیاه باشند و پیکسل‌هایی که احتمالاً نشانگر یک چهره هستند به رنگ سفید نمایش داده شود.

تا اینجا مرحله اول پروژه شما انجام شده است. برای آنکه نتیجه دقیق‌تری به دست بیاورید مراحل زیر را ادامه دهید.

۶- با توجه به مبحث شماره‌گذاری که در اسلایدهای توضیح آمده است، تعداد بخش‌هایی از تصویر را که ممکن است نشانگر یک چهره باشد، تعیین کنید. (مجموعه پیکسل‌های سفید متصل به هم را به عنوان یک شیء در نظر بگیرید - ربطی به مفهوم برنامه‌نویسی شیء‌گرا ندارد! معنی فارسی شیء منظور است!) - سپس تعداد اشیاء را بشمارید)

۷- تعداد چهره‌های بالقوه (تعداد اشایء مرحله قبل) را چاپ کنید. همچنین برای هر شیء مشخص کنید که اگر بخواهیم کادری دور آن در نظر بگیریم، طول، عرض و مساحت کادر چقدر خواهد بود.

مرحله دوم پروژه شما هم به پایان رسید. هم‌اکنون شما تعداد چهره‌های بالقوه را به دست آورده‌اید. برای آنکه ببینید از این میان کدام‌ها واقعا چهره هستند و کدام‌ها نه، لازم است مراحل زیر را دنبال کنید.

۸- تست‌های شماره ۴ تا ۶ موجود در اسلایدها را انجام دهید تا گزینه‌های غیرمرتبط را شناسایی کنید.

۹- دور گزینه‌های باقی‌مانده، کادری رسم کنید.

کار شما اینجا به پایان می‌رسد. تبریک می‌گوییم. شما یک برنامه **Face Detection** نوشته‌اید.

در صورت تمایل می‌توانید تست شماره ۷ اسلایدها را هم انجام دهید تا الگوریتم شما بهتر شود.