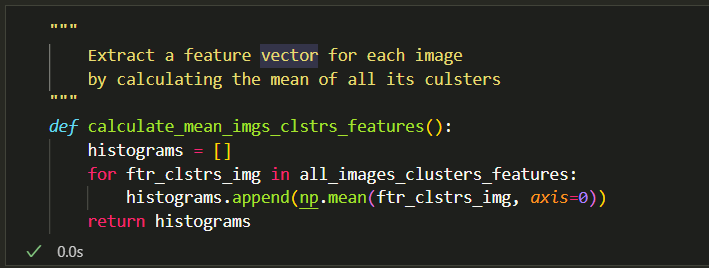
در این فاز از ما خواسته شده که مهم ترین نواحی را تشخیص دهیم و اهمیت آنها را مشخص کنیم.

باید نواحی ای را پیدا کنیم که بیشترین اطلاعات را درباره طبقه بندی تصویر به ما می دهند. برای طبقه بندی تصاویر پس از استخراج ویژگی از ناحیه ها، میانگین بردار ویژگی ناحیه ها را بدست میاوریم و از بردار حاصل به عنوان بردار ویژگی تصاویر استفاده میکنیم.

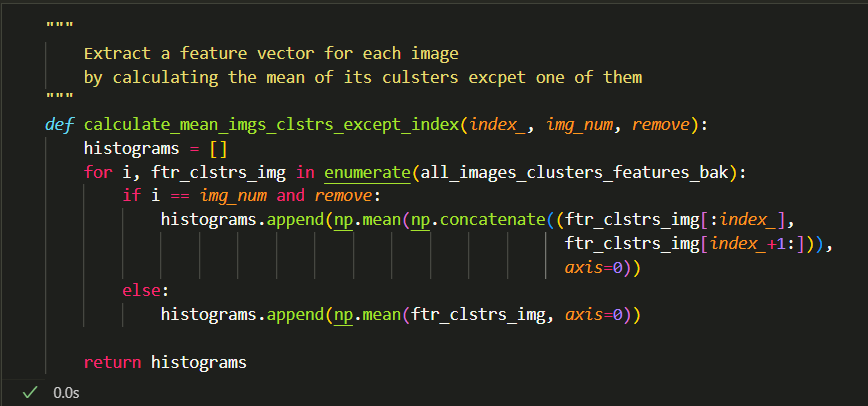
با استفاده از تابع زیر (calculate\_mean\_imgs\_clstrs\_features) میانگین بردار ویژگی کلاستر های هر عکس را میتوانیم بدست آوریم. Histograms در حقیقت لیستی حاوی بردار ویژگی تصاویر است.



از این تابع برای تست کردن درستی میانگین گیری استفاده می‌شد. آنرا توسعه دادیم و به تابع پایین تبدیل کردیم که آنرا در پایین توضیح داده ایم.

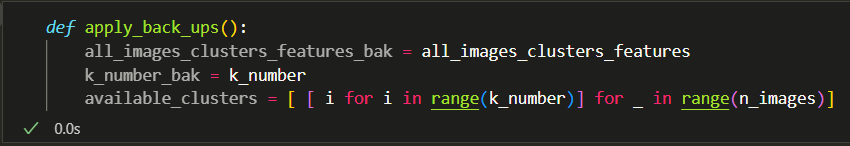
با استفاده از تابع زیر (calculate\_mean\_imgs\_clstrs\_except\_index) میتوانیم میانگین بردار ویژگی کلاستر های یک عکس را (img\_num) ، بجز یک کلاستر خاص(index\_) در آن عکس، بدست آوریم.

کاربرد این تابع آنجاست که میخواهیم به ازای هر عکس، با حذف هر کدام از کلاستر های آن، تغییرات احتمال لیبل آن عکس را بررسی کنیم که ببینیم بعد از حذف هر کدام از کلاستر های عکس، اطمینان لیبل آن چگونه تغییر میکند تا بر اساس آن ترتیب اهمیت کلاستر های عکس را مشخص کنیم.

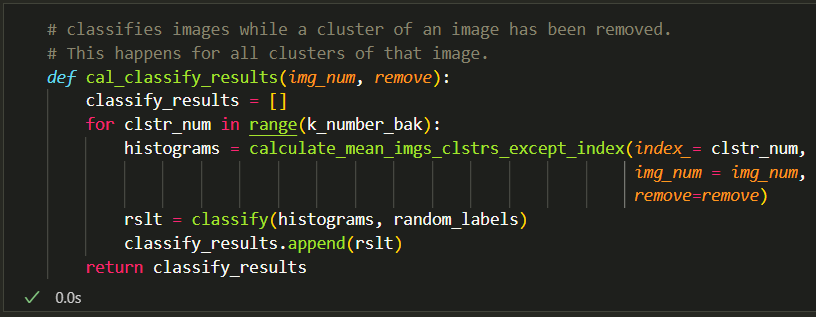


اینجا histograms لیستی شامل میانگین بردار ویژگی کلاستر های عکس هاست که یک کلاستر از یک عکس خاص آن حذف شده و در بردار ویژگی آن عکس اعمال نشده است.

در این قسمت کد برای لیست های کاربردی یک instance ایجاد کرده ایم و تغییرات را روی آنها اعمال میکنیم تا در هنگام تست کردن لیست های اصلی تغییر نکنند و نیاز به ران کردن مجدد کد از اول نباشد.

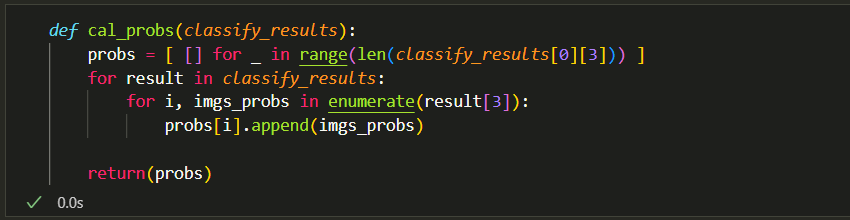


سپس با استفاده از تابع cal\_classify\_results میتوانیم به ازای حذف هر یک از کلاستر های یک عکسِimg\_num لیست histograms را بدست آوریم. سپس این histograms را به تابع classify می‌دهیم تا classification را انجام دهد. خروجی این تابع (rslt) شامل اطمینان لیبل عکس هاست که در توابع بعدی از آن استفاده میکنیم. (به تعداد k\_number\_bak که همان تعداد کلاستر های k-means است این کار را میکنیم.) (تابع classify را در انتهای داکیونت توضیح می‌دهیم)



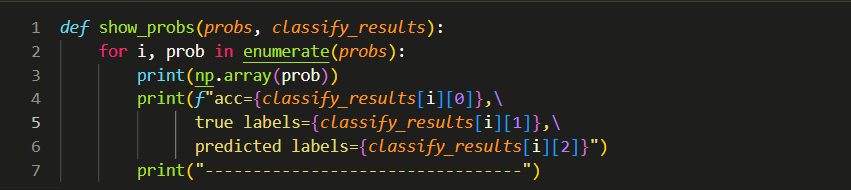
در نهایت classify\_results، یعنی خروجی همین تابع، به ازای هر عکس شامل چندین پارامتر است که یکی از آنها همان اطمینان لیبل عکس مورد نظر (img\_num) بعد از حذف هر یک از کلاستر های آن است.

سپس با استفاده از تابع cal\_probs (calculate probabilities) به ازای هر عکس اطمینان لیبل بعد از حذف هر یک از کلاستر های آن که در classify\_results ذخیره شده بود را جمع آوری کرده و در یک لیست کنار هم قرار میدهیم تا بتوان بعدا اطمینان لیبل هر عکس در نبود هر کلاستر های آنرا با هم مقایسه کرد و کلاستر ها را رتبه بندی کرد.

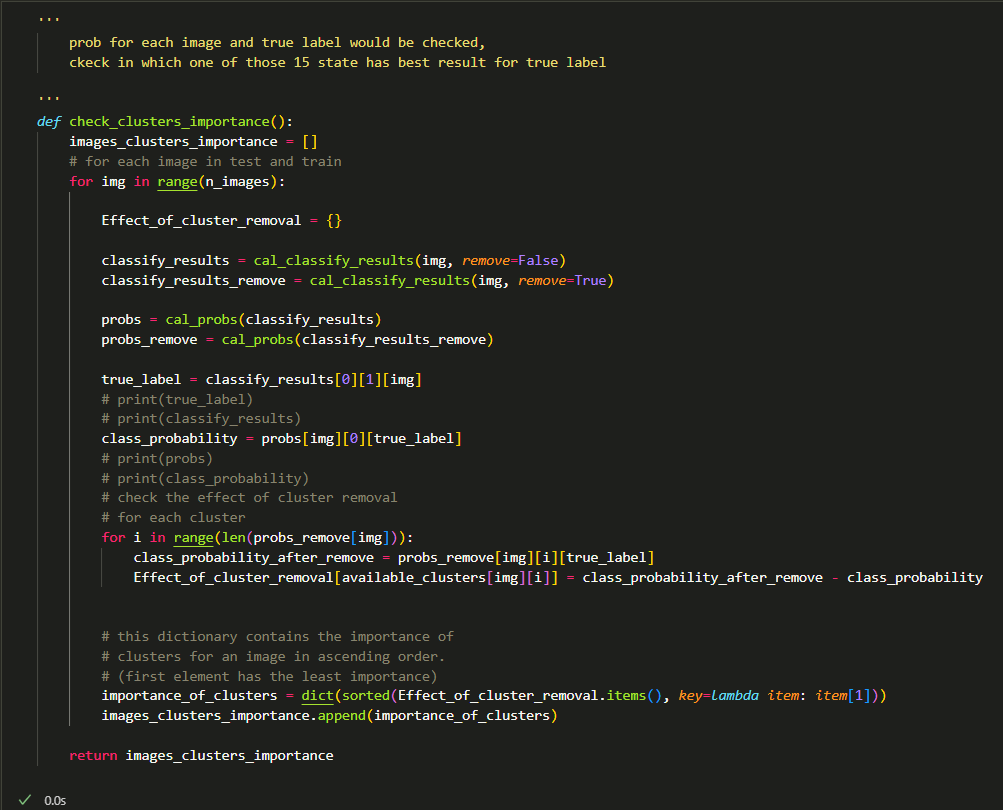


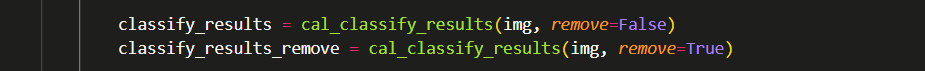
هر کدام از لیست های داخل لیست probs شامل اطمینان لیبل ها برای یک عکس در نبود هر یک از کلاستر های آن است.

از تابع زیر (show\_probs) برای تست کردن درستی حذف هر کلاستر و دیدن تاثیر آن در اطمینان لیبل عکس استفاده می‌شد.

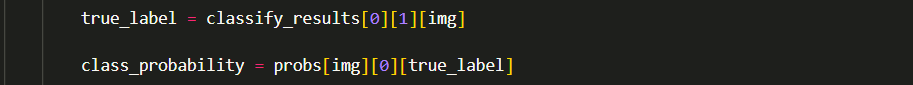


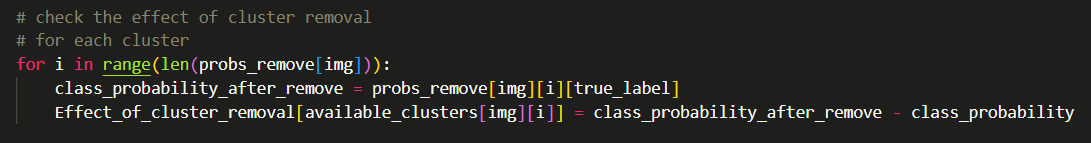
سپس با استفاده از تابع check\_clusters\_importance اهمیت هر یک از کلاستر های هر عکس و اولویت آنها را بدست می‌آوریم. در ادامه بخش های مختلف آن و روش کارش را توضیح می‌دهیم.

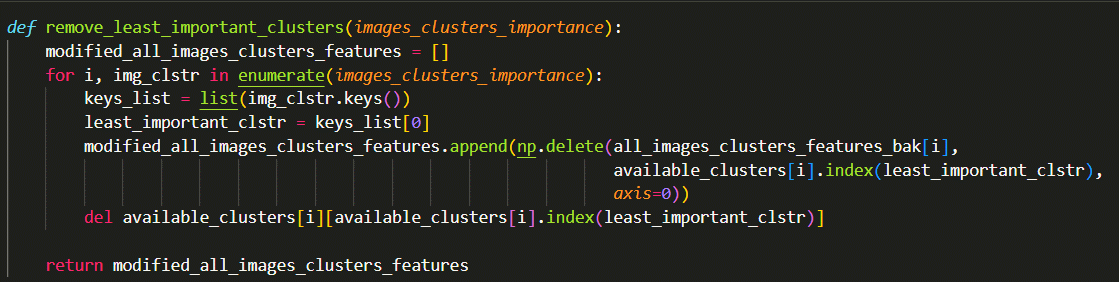


روش کار تابع بالا این صورت است که به ازای هر عکس، یک دیکشنری Effect\_of\_cluster\_removal در نظر میگیریم که در آن هر key نشان دهنده شماره کلاستر است و value آن نشان دهنده این است که بعد از حذف این کلاستر اطمینان عکس برای لیبل آن به چه صورت تغییر میکند.   
classify\_results خروجی تابع cal\_classify\_results است که نتایج classification را برای حالتی که کلاستری حذف نشده (remove=False) به ما میدهد.  
classify\_results\_remove خروجی تابع cal\_classify\_results است که نتایج classification را برای حالتی که کلاستر ها در آن عکس حذف شده باشند (remove=True) به ما میدهد.  
(نتایج classification همانطور که قبلا گفته شد شامل لیبل های صحیح عکس ها، لیبل های پیش بینی شده برای عکس ها، اطمینان هر لیبل برای هر عکس و accuracy است.)

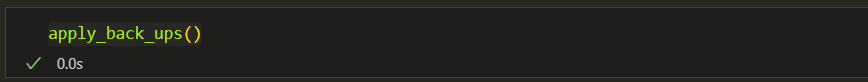
probs و probs\_remove به ترتیب حاوی اطمینان هر لیبل برای هر عکس در حالتی که کلاستر ها را حذف نکنیم و حذف کنیم است.

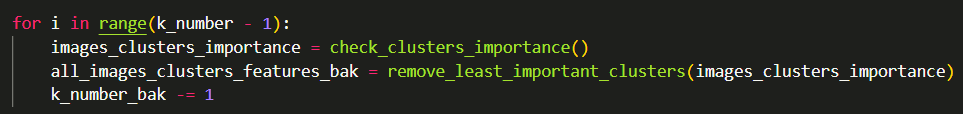
سپس true\_label یعنی همان لیبل اصلی عکس و class\_probability یعنی اطمینان لیبل اصلی برای عکس را بدست میاوریم.

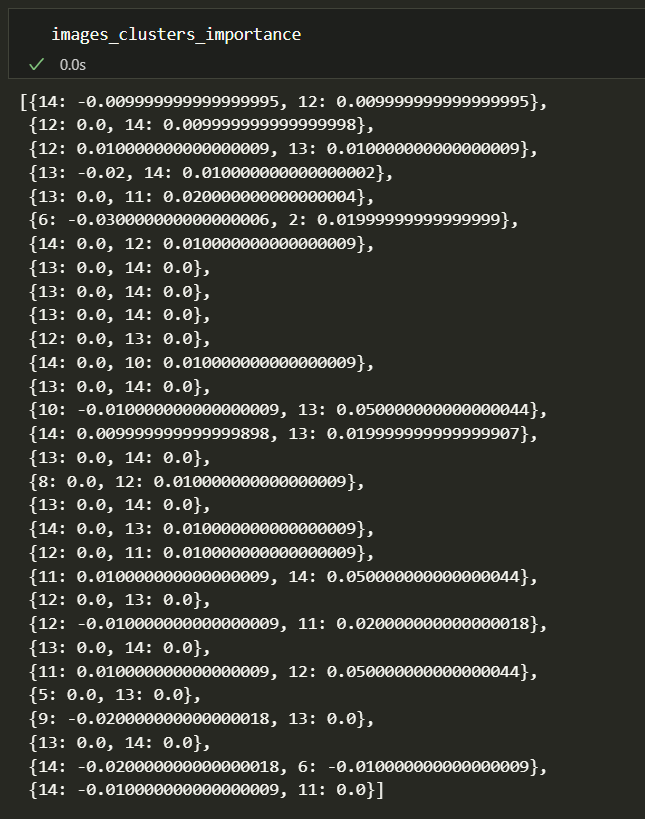
سپس به ازای هر کلاستر عکس، اطمینان لیبل عکس در نبود آن کلاستر را بدست میاوریم. همچنین Effect\_of\_cluster\_removal را نیز که بالاتر توضیح داده ایم برای آن کلاستر بدست میاوریم.  
در نهایت این دیکشنری را به صورت صعودی sort میکنیم و در importance\_of\_clusters ذخیره میکنیم (اولین عضو کمترین اولویت را دارد) و به لیست images\_clusters\_importance که شامل همه‌ی این دیکشنری ها برای همه‌ی عکس هاست اضافه می‌کنیم.

با استفاده از تابع remove\_least\_important\_clusters کم اهمیت ترین کلاستر هر عکس را از لیست کلاستر هایش که در فاز قبلی در لیست all\_images\_clusters\_features ذخیره کرده بودیم حذف میکنیم.

بنابراین لیست modified\_all\_images\_clusters\_features شامل بردار ویژگی همه کلاستر های همه عکس هاست که کم اهمیت ترین کلاستر آنها حذف شده است.

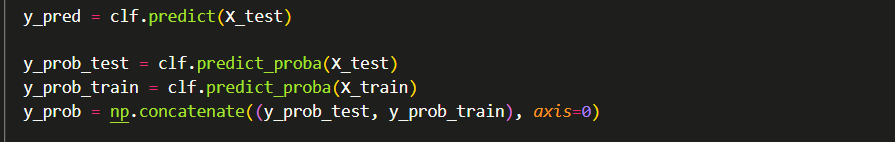
تابع apply\_back\_ups را قبل از شناسایی و حذف کلاسترهای کم اهمیت صدا میزنیم تا همانطور که گفته شد از لیست های اصلی یک instance بسازد و ما با آنها کار کنیم.

در این حلقه for به تعداد k\_number-1 بار توابع check\_clusters\_importance و remove\_least\_important\_clusters را صدا میزیم تا همانطور که توضیح داده شد، کم اهمیت ترین کلاستر ها شناسایی و حذف شوند.

سپس کلاستر های مهم هر عکس را به صورت زیر نمایش میدهیم: (هر دیکشنری نماینده یک عکس است و مقدار key آن شماره‌ی مهمترین کلاستر و مقدار value تغییر میزان اطمینان لیبل عکسش بعد از حذف آن کلاستر است)

دو کلاستر مهم هر عکس بدست آمده است. اندیس 1 اولین مهمترین و اندیس صفر دومین مهمترین کلاستر است.

تابع classify که پیشتر از خروجی آن استفاده کرده بودیم نیز به این صورت عمل میکند که پس از fit و predict کردن داده ها، لیبل های پیشبینی شده برای داده های تست را در لیست y\_pred و میزان اطمینان هر لیبل برای عکس های train و test را در آرایه y\_prob ذخیره میکنیم.

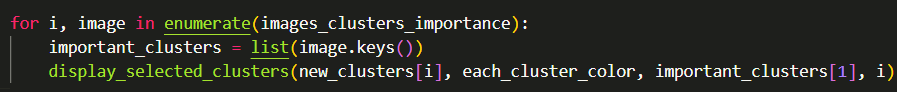


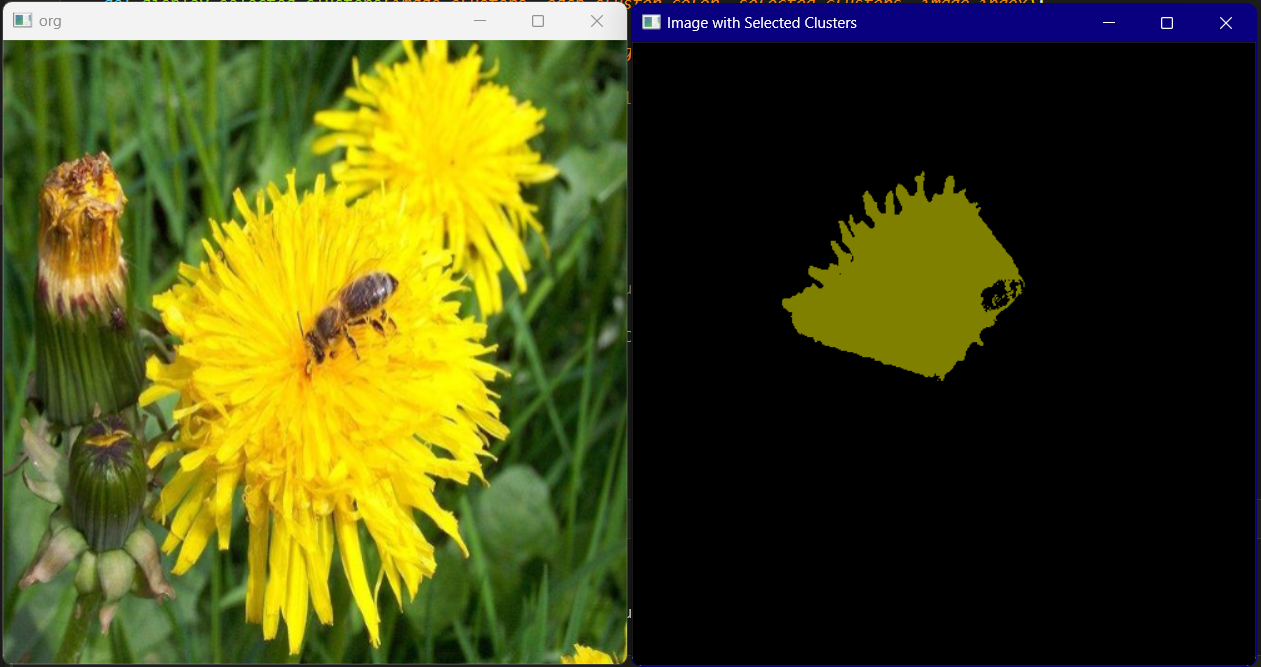
همچنین لیبل درست داده ها را در لیست true\_labels ذخیره میکنیم. accuracy را نیز بدست میاوریم.   
در انتها این مقادیر را return میکنیم و همانطور که گفته شد در مواقع نیاز استفاده میکنیم.

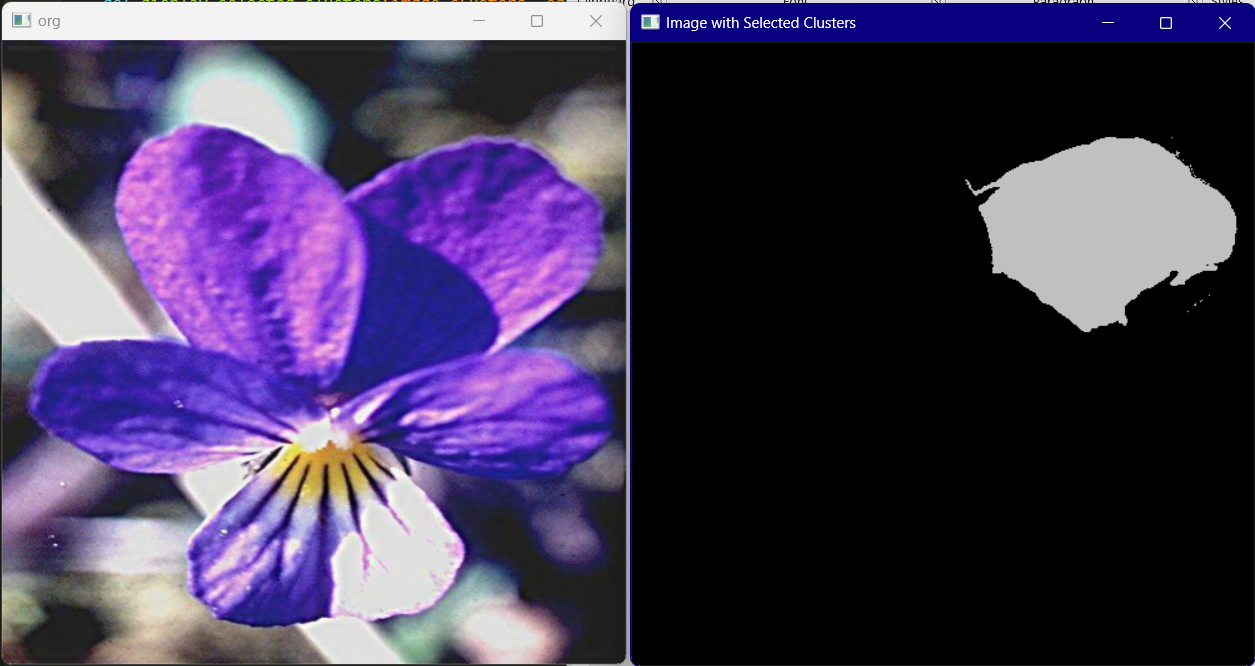


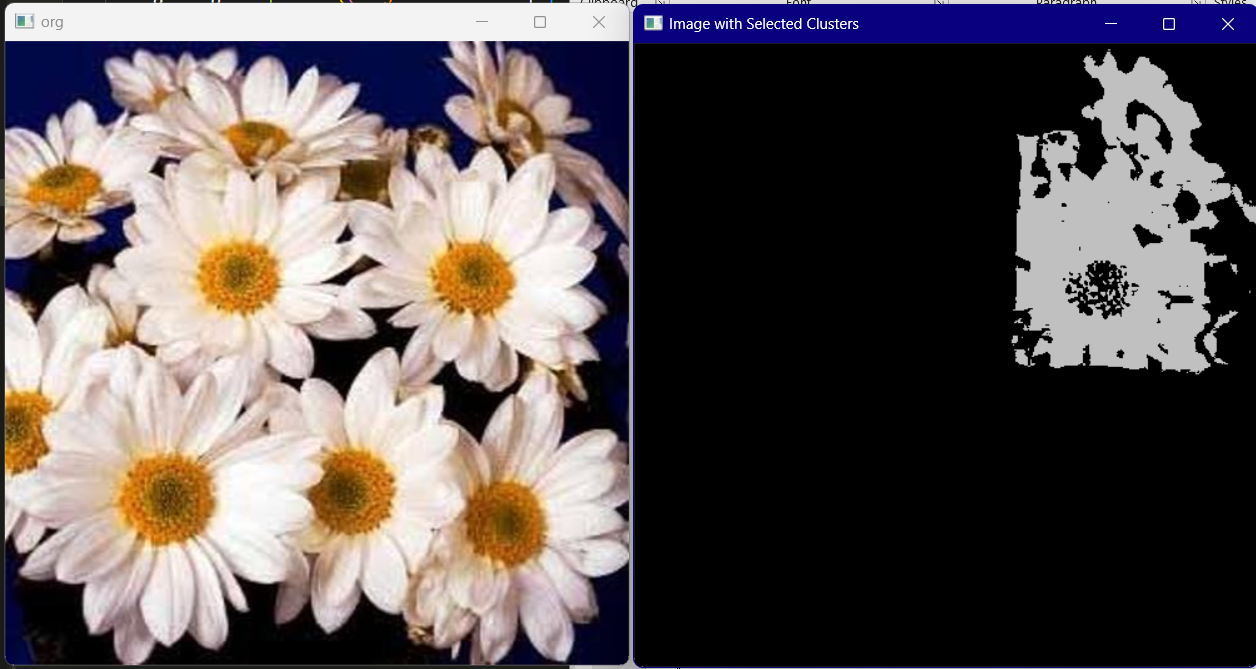
در نهایت با استفاده از تابع display\_selected\_clusters مهمترین کلاستر هر عکس را نمایش میدهیم.

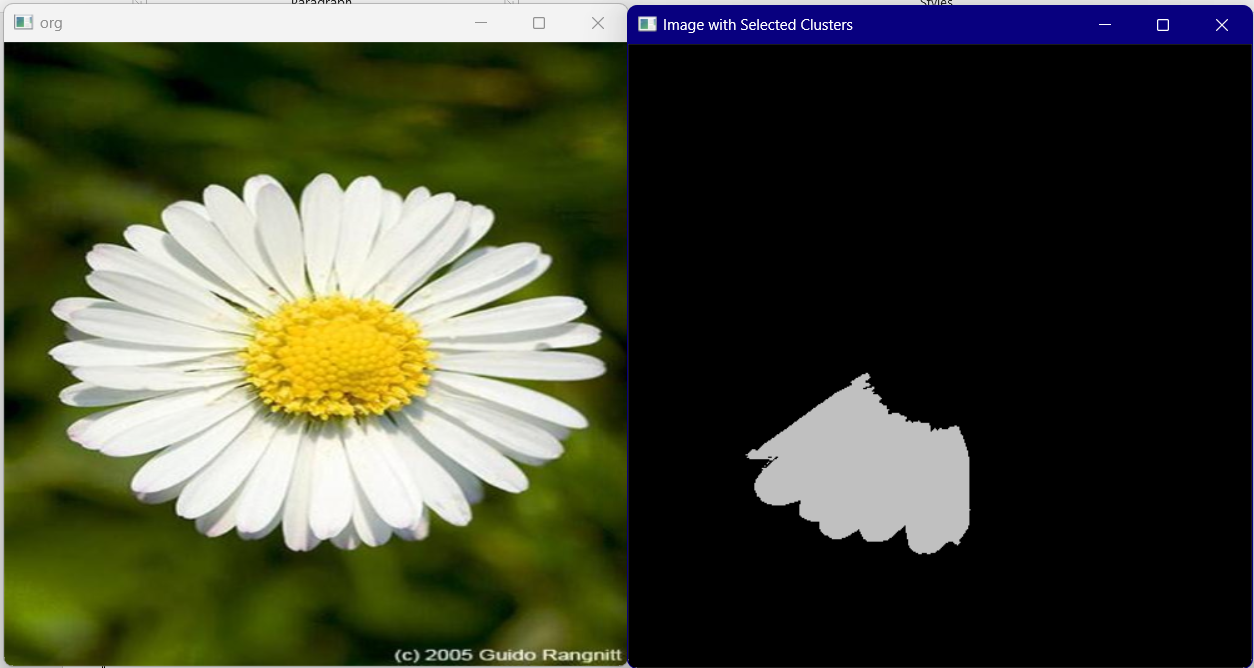




خروجی این تابع، یعنی مهمترین کلاستر هر عکس، را برای نمونه در اینجا نشان میدهیم:  
به دلیل اینکه تعداد کلاستر های هر تصویر 15 عدد است، مهمترین کلاستر بخشی از گل است نه همه‌ی آن.







در بیشتر موارد مهمترین کلاستر به درستی پیدا می‌شود، ولی در چند مورد ممکن است تشخیص صحیح نباشد، مانند نمونه زیر:

