

به نام خدا

گزارش پروژه پایانی درس بینایی ماشین

در این پروژه بر روی روش ها و ایده های بهینه تر کردن روش seam carving کار میکنیم. به نظر من میتوانیم بهینه کردن روش seam carving را در دو مرحله بررسی کنیم.

۱ - مرحله ایجاد و تولید energy map تصویر

۲ - مرحله چگونگی یافتن بهینه ترین seam از بین تمام seam های موجود

تغییرات مرحله اول:

ابتدا برای یافتن انرژی مپ تصویر ورودی تنها از یک گرادیان ساده استفاده کردیم که برای برخی تصاویر مانند تصویر زیر خروجی مناسبی میداد. اما باعث خرابی بعضی تصاویر دیگر میشد. مانند تصویر Diana موجود در زیر:





روش های مختلفی برای یافتن یک انرژی مپ مناسب که بتواند آبجکت های مهم تصویر را تشخیص دهد وجود دارد مانند یافتن عمق تصویر یا ... اما در هر کدام از آنجایی که انسان تشخیص نمی دهد و بر اساس رنگ و روشنایی تصویر تشخیص داده میشود، ممکن است خطاهای زیادی داشته باشد و برای هر تصویری پاسخگو نباشد.

پس برای اینکه بتوانیم هر چه دقیق تر اجزای مهم تصویر را تشخیص دهیم (و از آنجایی که شرط خاصی برای پروژه قرار داده نشده است)، می خواهیم از انسان کمک بگیریم تا تشخیص های نادرست هر چه بیشتر کاهش یابد. لذا در اینجا علاوه بر استفاده از گرادیان، عمق تصویر و برجستگی ها، همین پارامتر ها را برای قطعات کوچکتر و مهمتر تصویر نیز که توسط کاربر انتخاب شده و جدا سازی شده است را اضافه میکنیم.

برای راحتی این کار یک اسکریپت که فایل آن کنار سایر فایل ها قرار دارد نوشته شده است که تصویر ورودی را به تعداد دلخواه بخش مساوی تقسیم می کنید و در مسیر گفته شده در پایین میریزد و فقط کافی است که از بین این قطعات، آنهایی که از اهمیت کمی برخوردار هستند را پاک کنیم.

`./helper/pic_name/`

بدیهی است که هر چه تصویر اصلی را به تعداد سگمنت های کوچک تری تبدیل کنیم و تصاویر کمی نهایی تعداد بیشتر و کوچک تر باشد، نتیجه نهایی بهتر خواهد شد.

تصاویری که در این پوشه قرار میگیرد و به کد داده میشود با استفاده از object detection تشخیص داده میشود که هر کدام از قطعات مربوط به کدام بخش تصویر است و انرژی مربوط به آن بخش نسبت به سایر بخش ها افزایش میابد.

حال چگونه این انرژی افزایش می یابد؟

ابتدا یک `base_energy` برای تصویر اصلی می یابیم. این کار را با جمع وزن دار سه پارامتر عمق، برجستگی و گرادیان انجام میدهیم و با آزمون و خطا به این مقدار میرسیم.

$$\text{base_energy} = 0.05 * \text{gradient_energy_map} + 0.15 * \text{smap} + 0.8 * \text{dmap};$$

یک تابع `energy_screan_shots` داریم که وظیفه این را دارد که انرژی قطعات تصاویری که به عنوان سگمنت های مهم به آن داده میشود را تشدید کند.

در پارامتر های اولیه کد یک متغیر به نام `ratio` مقدار دهی میشود که نشان دهنده میزان تاثیر این سگمنت های مهم بر روی انرژی مپ است. مثلا اگر این مقدار برابر ۲ باشد به این معنی است که انرژی مربوط به سگمنت های داده شده به کد را دوبرابر میکند. (به صورت جداگانه هم میتوانیم این ضریب را برای تک تک سگمنت ها در نظر گرفت مثلا در نام هر سگمنت یک عدد با همین مفهوم قرار داد تا بتوانیم به هر سگمنت وزن جداگانه و مشخصی بر اساس اهمیت آن بدهیم. که این کار در این ورژن از پروژه انجام نشده است:)

تغییرات مرحله دوم:

در مرحله دومی که می توانیم الگوریتم را بهینه کنیم این است که بتوانیم یک سیم بهینه و مناسب برای حذف آن پیدا کنیم. روش پیاده سازی این قسمت به همان صورت است که در کلاس توضیح داده شد و با استفاده از `dp` مسیری که مجموع انرژی کمتری داشته باشد را انتخاب و حذف میکنیم. البته یک پارامتر برای این بخش از ورودی میگیریم به نام `each_level_seam_carving_look_ahead` که یک بازه برای انتخاب پیکسل در هر مرحله در فرایند تصمیم گیری انتخاب پیکسل بعدی است. الگوریتمی که در کلاس توضیح داده شد به این صورت بود که هر پیکسل برای مرحله بعدی تنها میتواند پیکسل هم ستون خود و یا چپ یا راست آن را انتخاب کند و پیش برود. اما ما در اینجا با دریافت این پارامتر، این اختیار را به الگوریتم میدهیم که این تعداد پیکسل میتواند از چپ و راست برای انتخاب پیکسل بعدی در نظر بگیرد. یعنی مثلا اگر مقدار آن برابر ۵ باشد، در انتخاب پیکسل در هر مرحله علاوه بر اینکه پیکسل هم ستون خود را چک میکند، میتواند ۵ تا چپ یا ۵ تا راست را نیز چک کند و بهترین مسیر را انتخاب کند. که البته بدیهی است که اگر مقدار آن زیاد باشد باعث ایجاد بریدگی هایی در تصویر نهایی میشود که با چشم قابل تشخیص خواهد شد.

تصاویر خروجی زیر با پارامتر های ذکر شده بدست آمده است:

Tower :

بدون تصویر کمکی و تنها با استفاده از گرادیان

`each_level_seam_carving_look_ahead = 1;`

Before:



After 50% resize:



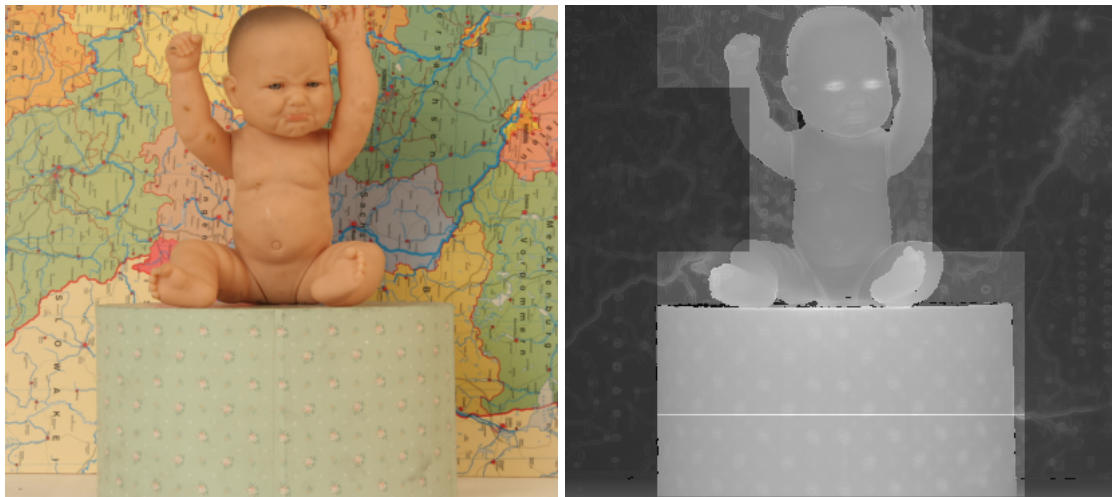
Baby:

سگمنت کردن تصویر به ۳۶ قطعه کوچک تر و انتخاب ۱۹ تا از مهم ترین های آن.

Ratio = 2

each_level_seam_carving_look_ahead = 1;

Before:



After 50% resize:



Diana:

سگمنت کردن تصویر به ۳۶ قطعه کوچک تر و انتخاب 21 تا از مهم ترین های آن.

Ratio = 2

each_level_seam_carving_look_ahead = 1;

Before:



After 50% resize:



Snowman:

سگمنت کردن تصویر به 64 قطعه کوچک تر و انتخاب 26 تا از مهم ترین های آن.

Ratio = 2

each_level_seam_carving_look_ahead = 1

Before:



After 50% resize:



Dolls:

سگمنت کردن تصویر به 64 قطعه کوچک تر و انتخاب 43 تا از مهم ترین های آن.

Ratio = 3

each_level_seam_carving_look_ahead = 1;

Before:



After 50%:

