

## گزارشکار تسک سوم دوره هوش مصنوعی تک استک ۲۰۲۵

### • بخش اول:

در ابتدا فیلم آموزشی قرار داده شده در یوتوب رو دیدم و از اونجایی که یه سری ابهام برام باقی مونده بود سراغ یار همیشگیمون هوش مصنوعی رفتم و با یکم صحبت متوجه شدم که دقیقا الگوریتم به چه شکل هست مثلا روند دقیق آپدیت نقاط مرکزی اولیه و ...

تا حد امکان کد به صورت بهینه بدون حلقه اضافه و اشغال حافظه غیر ضروری نوشته شده و از ابزارهای `numpy` به خاطر سرعت خوب استفاده شده به طور مثال مستقیما از تابع `norm` کتابخونه `numpy` استفاده شده. (همچنین در بخش دوم نگاه کلی به عکس وارد شده به این شکل هست که اون رو دنباله‌ای از پیکسل‌ها میبینیم تا عملیات‌هایی مثل انتخاب نقاط رندوم اولیه روند ساده‌تری داشته باشد همچنین باید به نکاتی مثل اینکه نقطه رندوم تکراری نداشته باشیم توجه کرد.

همیشه سعی می‌شود تا حد امکان ساختار کد اولیه تسک‌ها حفظ شود تا فرایند چک کردن عملکرد برای منتورهای گرامی راحت‌تر باشد به همین دلیل از موارد کوچکی مثل افزایش مصرف حافظه با لاگ کردن رشته‌ها جای اعداد صرف نظر شده.

به سراغ راه‌های بهبود نفرتم اما به نظر می‌رسد اگر نقاط اولیه را بتوان دقیق‌تر انتخاب کرد تعداد تکرار بسیار کمتری برای الگوریتم نیاز باشد به طور مثال وقتی با عکس `lena` و تکرار ۵ بار برای ۱۶ رنگ چند بار کد رو ران کردم اکثر اوقات مقدار نرم نهایی حدودا برابر ۳۴ بود ولی بعضا در یک اجرا به ۳۰ می‌رسید که مشخصا اهمیت انتخاب نقاط اولیه را نشان می‌دهد.

در نهایت در مورد زمان اجرای کد از آنجایی که همه چیز بهینه و با ابزارهای درست پیاده سازی شده این سوال پیش میاد که پس نرم افزارهای فشرده سازی عکس چطور این فرایند را در کمتر از یک ثانیه در مقایسه با زمان چند ثانیه‌ای این کد انجام می‌دهند. آیا صرفا تفاوت زبان برنامه‌نویسی است یا استفاده از تکنیک خاص، موازی سازی یا کلا الگوریتم بهینه تر.

از دیگر کارهای قابل انجام ذخیره عکس نهایی در فرمت مناسب جهت مقایسه واقعی حجم عکس است چرا که فرمت `png` یک ذخیره سازی پیکسلی انجام می‌دهد و تفاوتی در حجم عکس ایجاد نمی‌شود.

از جالب‌ترین نتایج نهایی محدود کردن تعداد رنگ به ۲ یا ۳ عدد می‌باشد که بسیار شبیه برخی فیلترهای زیبایی عکس نرم افزارها و حالت نقاشی طور می‌شود که کاربردهای دیگر این الگوریتم را نشان می‌دهد.

## • بخش دوم:

مجدد فیلم آموزشی رو در یوتوب دیدم که تقریبا هر چیزی که نیاز بود داخلش بررسی شده بود و مشکل خاصی نداشتم. از اونجایی که دو نقشه مختلف در دسترس بود و نقشه گرادیان هم به راحتی قابل محاسبه بود به فکر ترکیب نقشه ها بودم که ایده ام با یک ایده خوب از هوش مصنوعی یعنی وزن دار کردن نقشه ها کامل شد و نتیجه با کمی آزمون و خطا و پلات کردن نقشه نهایی و بررسی بصری آن بسیار مناسب شد طوری که با تقسیم وزن مناسب نقشه نهایی به میزان قابل توجهی سوژه اصلی را بولد میکند و به الگوریتم کمک میکند همچنین این امکان را فراهم میکند که با وزن صفر دادن برای نقشه ها کامل آن را کنار بگذاریم و بررسی کنیم.

اگرچه این وزن ها داخل تابع هاردکد شدن اما ساختار کد به شدت ماژولار و قابل شخصی سازی است و هدف باز هم بیشتر حفظ ساختار کد اولیه تسک برای صدا زدن تابع بوده.

روند کلی برنامه شکسته شدن هر بخش به یک تابع مجزاس که برای خلاصه ماندن گزارش صرفا معرفی می شوند و توضیحات تکمیلی در کامنت های کد آورده شده.

در فایل نوت بوک از آخر به اول به ترتیب تابع اصلی رو داریم که مدیریت روند کد رو در اختیار داره.

تابع ساخت نقشه گرادیان رو داریم که با فیلتر **sobel** محاسبه می شود.

تابع تولید یک نقشه نهایی بر اساس نقشه ها و وزن های دلخواه رو داریم.

تابع تولید ماتریس انرژی که از ایده های قشنگ مورد نیاز این تسک هست که از بالا به پایین محاسبه می شود.

تابع پیدا کردن مسیر با کمترین انرژی رو داریم که صرفا مسیر رو برامون مشخص می کنه.

تابع نمایش مسیر روی عکس و حذف اون مسیر از عکس و نقشه های ممکن رو داریم.

در همین ساختار ماژولار نیز تمامی توابع انعطاف پذیری های زیادی رو در خود جای دادن و سناریوهای مختلف و قابلیت های مختلف مثلا پلات کردن عملکرد همان تابع برایشان در نظر گرفته شده.

در جهت نمایش انیمیشن حذف مسیرها در پنجره جدا دستور مربوطه در خط آخر سلول اول قرار داده شده.

ایده تغییر سایز در ارتفاع بسیار سادس فقط کافیه عکس ۹۰ درجه چرخیده بشه و همین روند فعلی انجام بشه در نتیجه از پیاده سازی آن صرف نظر شد :)

در نهایت خروجی از لحاظ بصری روی هر سه مورد تست کیس در دسترس، بسیار مناسب و سوژه اصلی کاملا حفظ شده.

<>> با تشکر از این تسک جذاب <<<