

باسمه تعالی

تکلیف سری پنجم درس سیستم های چندرسانه ای

سارا برادران (شماره دانشجویی: ۹۶۲۴۱۹۳)

بلوک [1] فایل ipynb : تغییر backend به Tkinter

The `ginput()` function is not available in the inline mode. If you need it, you should start the notebook using some other backend that you have installed. For example `qt` or `tk`:

Tkinter is a standard library in Python which is used for GUI application. Tkinter has various controls which are used to build a GUI-based application.

```
pip install tk
```

بلوک [2] فایل ipynb : کتابخانه ها

در این قسمت کتابخانه های به کار رفته در کد `import` شده است. به طور کلی از ۳ کتابخانه `numpy`، `cv2` و `matplotlib` استفاده نموده ایم که نصب هر یک از این کتابخانه ها به کمک دستورات زیر قابل انجام است. کتابخانه `matplotlib` برای نمایش تصاویر، کتابخانه `cv2` برای اعمالی از جمله خواندن تصاویر و کتابخانه `numpy` برای انجام برخی عملیات ها بر روی تصاویر مورد استفاده قرار گرفته است که در ادامه به تفصیل به آن ها می پردازیم.

```
pip install numpy
pip install matplotlib
pip install opencv-python
```

بلوک [3] فایل ipynb : تابع `Show_Images()`

در ابتدا یک تابع تحت عنوان `show_images` برای نمایش تصاویر به صورت تکی و چندتایی ایجاد شده است. برای نمایش تصویر و پیاده سازی این تابع از کتابخانه `matplotlib` و دستور `imshow` استفاده کرده ایم. همچنین این تابع به عنوان آرگومان ورودی لیستی از تصاویر، برچسب هر تصویر، و سایز مورد نیاز برای نمایش تصاویر را دریافت می نماید. به علاوه این تابع ابعاد تصاویر دریافتی را در کنار برچسب نام هر تصویر نمایش می دهد. از تابع پیاده سازی شده در مراحل بعدی و برای نمایش تصویر خروجی حاصل از توابع پیاده سازی شده استفاده می کنیم.

بلوک [4] فایل ipynb : خواندن تصویر ورودی و نمایش آن

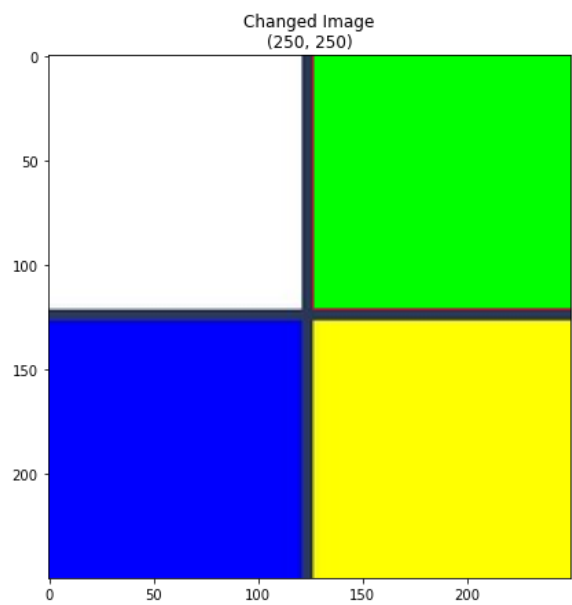
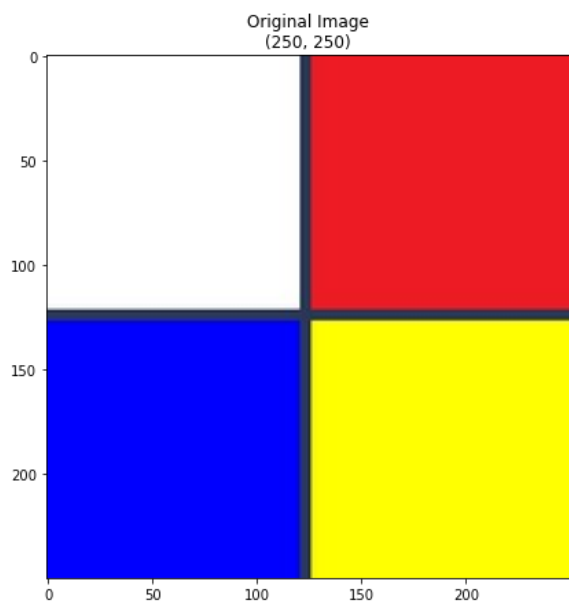
در این قسمت ابتدا به وسیله تابع `imread` کتابخانه `cv2` تصویر `pic.bmp` را خوانده و درون `src_img` ذخیره می نماییم. با توجه به رنگی بودن تصویر از متد `COLOR_BGR2RGB` برا خواندن آن استفاده می نماییم.

بلوک [5] فایل ipynb : تابع `Color_Change()`

در این قسمت تابعی تحت عنوان `Color_Change` پیاده سازی شده است که تصویری را به عنوان ورودی دریافت می کند. به علاوه این تابع مختصات یک پیکسل، یک آستانه تحت عنوان `w` و نیز یک رنگ جدید را در آرگومان های ورودی دریافت می نماید. سپس در بدنه تابع ابتدا ابعاد تصویر درون متغیر های `height` و `width` ذخیره شده و یک کپی از تصویر ورودی درون `new_img` قرار می گیرد. سپس پیکسل انتخاب شده درون `sample_point` ذخیره شده و با پیمایش بر روی تمامی پیکسل های تصویر اولیه در صورتی که حداقل یک پیکسل مقدار یکی از رنگ های `RGB` آن بیش از `w` با رنگ متناظر در `sample_point` اختلاف داشته باشد `flag` برابر `False` شده و در غیر اینصورت `flag` برابر `True` باقی مانده و لذا در انتها پیکسل هایی که `flag` نظیر آن ها `True` مانده است با رنگ جدید جایگزین می شود. (به این معنی که هر سه کانال رنگ آن ها با `sample_point` بیش از `w` اختلاف نداشته است.) تصویر تغییر داده شده در خروجی تابع بازگردانده می شود.

بلوک [6] فایل ipynb : فراخوانی تابع `Color_Change()` و نمایش تصویر خروجی

در این قسمت ابتدا تصویر اولیه نمایش داده شده و با استفاده از متد `ginput()` با کلیک بر روی یک نقطه از تصویر مختصات آن درون `in_point` ذخیره می گردد. سپس مختصات بدست آمده که در قالب یک لیست است به `tuple` تبدیل شده و دو عضو آن معکوس می گردد (در کار با تصاویر محور `x` و `y` معمولاً برعکس می باشد.) در انتها تابع `Color_Change` فراخوانی می شود و به عنوان آرگومان های ورودی تصویر اولیه `src_img`، به عنوان آستانه `w=50`، به عنوان پیکسل گزینش شده `in_point` و به عنوان رنگ جدید رنگ `[0, 255, 0]` (رنگ سبز) داده می شود حال می توان به خوبی مشاهده نمود که در تصویر حاصل از خروجی تابع تمام نقاط هم رنگ با نقطه انتخاب شده با ماوس به رنگ سبز تغییر کرده است. برای مثال در شکل (۱) نقطه با مختصات `(44,188)` گزینش شده بود که در محدوده ی قرمز تصویر بود و لذا تصویر نهایی کلیه پیکسل های محدوده قرمز را به رنگ سبز تغییر داده است. (برای تغییر `backend` به حالت اولیه برای چاپ تصویر در فایل `notebook` مجدداً از مد `inline` استفاده کرده و با استفاده از `%pylab inline` به مد اولیه باز میگردیم.



شکل (۱)