

باسمه تعالی

تکلیف سری چهارم درس سیستم های چندرسانه ای

سارا برادران (شماره دانشجویی: ۹۶۲۴۱۹۳)

بلوک [1] فایل ipynb : کتابخانه ها

در این قسمت کتابخانه های به کار رفته در کد import شده است. به طور کلی از ۳ کتابخانه cv2, numpy 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 [2] 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 سائز 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 [3] فایل ipynb : خواندن تصاویر و نمایش آنها

در این قسمت ابتدا به وسیله تابع imread کتابخانه cv2 تصویر IUT.bmp را خوانده و درون src_img ذخیره می نماییم. با توجه به سیاه و سفید بودن تصویر از متد COLOR_BGR2GRAY برا خواندن آن استفاده می نماییم.

بلوک [4] فایل ipynb : تابع H4_Shadow()

در این قسمت تابعی تحت عنوان H4_Shadow پیاده سازی شده است که تصویری را به عنوان ورودی دریافت می کند. ابتدا توسط تابع GaussianBlur کتابخانه cv2 تصویر به وسیله فیلتر گوسی با انحراف معیار d نرم شده و سپس تصویر نرم شده به اندازه S پیکسل به سمت پایین و سپس به اندازه S پیکسل به سمت راست شیفت داده می شود سپس با استفاده از مینیمم گیری میان هر یک از پیکسل های تصویر اولیه و تصویر نرم شیفت داده شده، تصویر حاصله که همان تصویر سایه دار است بدست خواهد آمد.

بلوک [5] فایل ipynb : فراخوانی تابع H4_Shadow() و نمایش تصویر خروجی

در این قسمت تابع H4_Shadow فراخوانی شده و `src_img` به عنوان تصویر اولیه و مقادیر $s=10$ و $d=10$ به عنوان مقدار شیفت و انحراف از معیار فیلتر گوسی تولید کننده سایه به این تابع پاس داده می شوند سپس تصویر اولیه و تصویر حاصل از خروجی تابع نمایش داده شده که مطابق شکل (۱) می باشد.



شکل (۱)

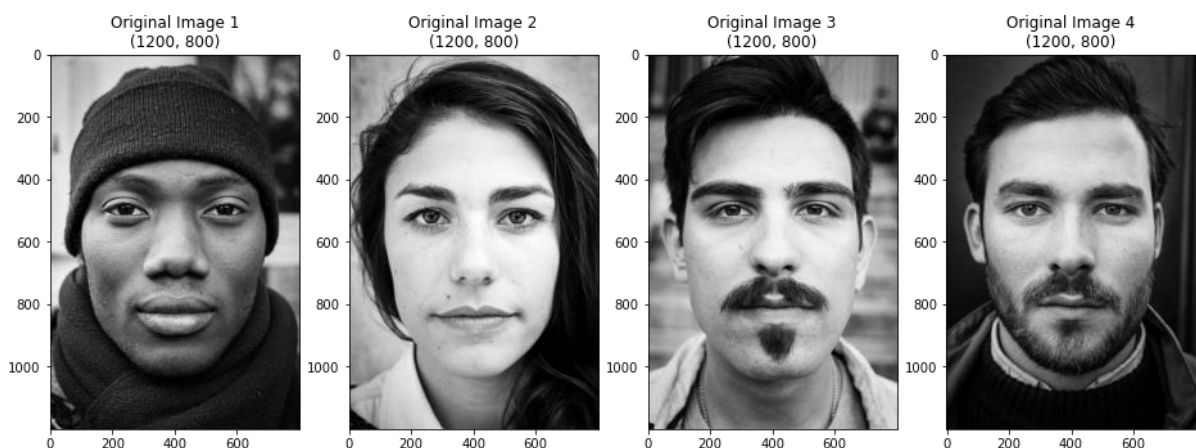
بلوک [6] فایل ipynb : تابع H4_Hybrid()

در این قسمت تابعی تحت عنوان H4_Hybrid پیاده سازی شده است که دو تصویر `src_img1` و `src_img2` را به عنوان تصاویر ورودی اولیه و مقادیر m و n را به عنوان انحراف از معیار فیلتر گوسی ای که می بایست بر روی دو تصویر اعمال شود، دریافت می کند. سپس تصاویر نرم شده پس از اعمال فیلتر گوسی بر روی دو تصویر را بدست آورده و در `blured_img1` و `blured_img2` ذخیره می نماید. همچنین تصاویر با نوع داده `uint8` به تصاویر با نوع داده `int` تبدیل شده اند تا در ادامه ضمن انجام عملیات های ریاضی بر روی تصاویر با سرریز مواجه نشویم. سپس تصویر فرکانس بالای `hfr_img` را به گونه ای تشکیل می دهیم

که تفاضل دو تصویر `src_img2` و `blured_img2` (تصویر حاصل از اعمال فیلتر گوسی بر روی `src_img2` باشد و در ادامه تصویر فرکانس بالای بدست آمده را با تصویر `blured_img1` (تصویر حاصل از اعمال فیلتر گوسی بر روی `src_img1`) جمع زده و این حاصل بدست آمده در حقیقت تصویر حاصل از ترکیب دو تصویر اولیه می باشد. حال با توجه به اینکه ضمن انجام عملیات های ریاضی فوق بر روی پیکسل های تصاویر ممکن است برخی از پیکسل های تصویر ترکیبی مقادیری بیش از ۲۵۵ یا کمتر از ۰ اخذ کرده باشند لذا یک مرتبه تمام پیکسل ها را مورد بررسی قرار داده و مقادیر اینگونه پیکسل ها را به مقادیر ۲۵۵ و ۰ تغییر می دهیم. در انتها نیز تصویر حاصل از ترکیب مجدداً به نوع داده `uint8` تبدیل شده و ماتریس تصویر نهایی به وسیله تابع `np.array()` به فرمت `numpy array` تبدیل می شود.

بلوک [7] فایل `ipynb`: خواندن تصاویر چهره و نمایش آنها

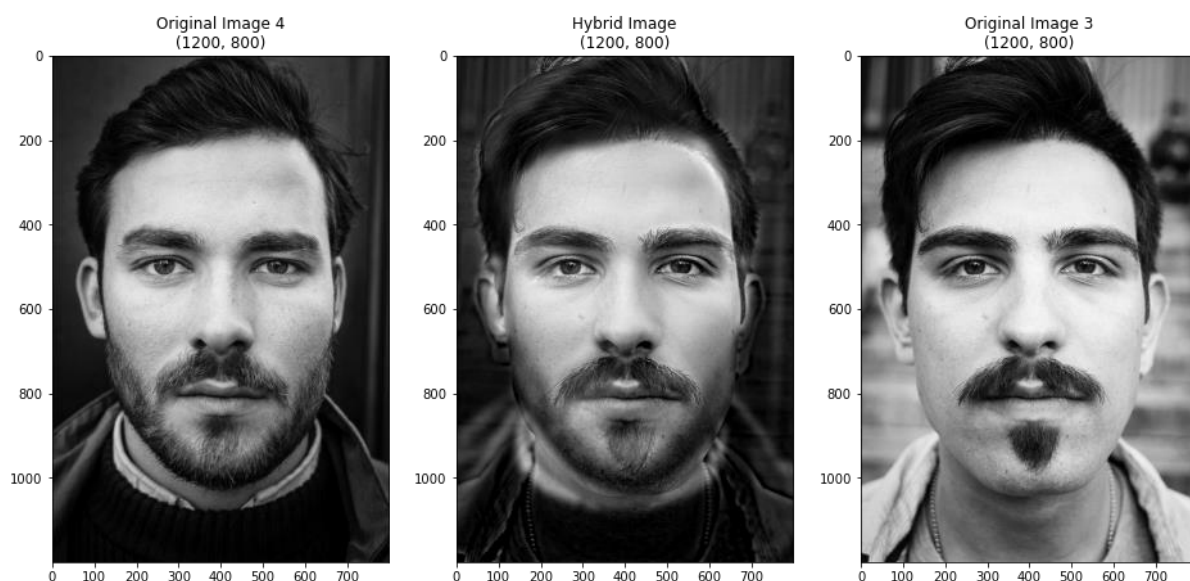
در این قسمت ابتدا به وسیله تابع `imread` کتابخانه `cv2` تصاویر چندین چهره که همگی طول و عرض یکسان دارند را خوانده و درون `src_img[1]` تا `src_img[4]` ذخیره کرده و توسط تابع `Show_Images` نمایش می دهیم که مطابق شکل (۲) می باشد.



شکل (۲)

بلوک [8] فایل `ipynb`: فراخوانی تابع `H4_Hybrid()` و نمایش تصویر خروجی و تصاویر اولیه

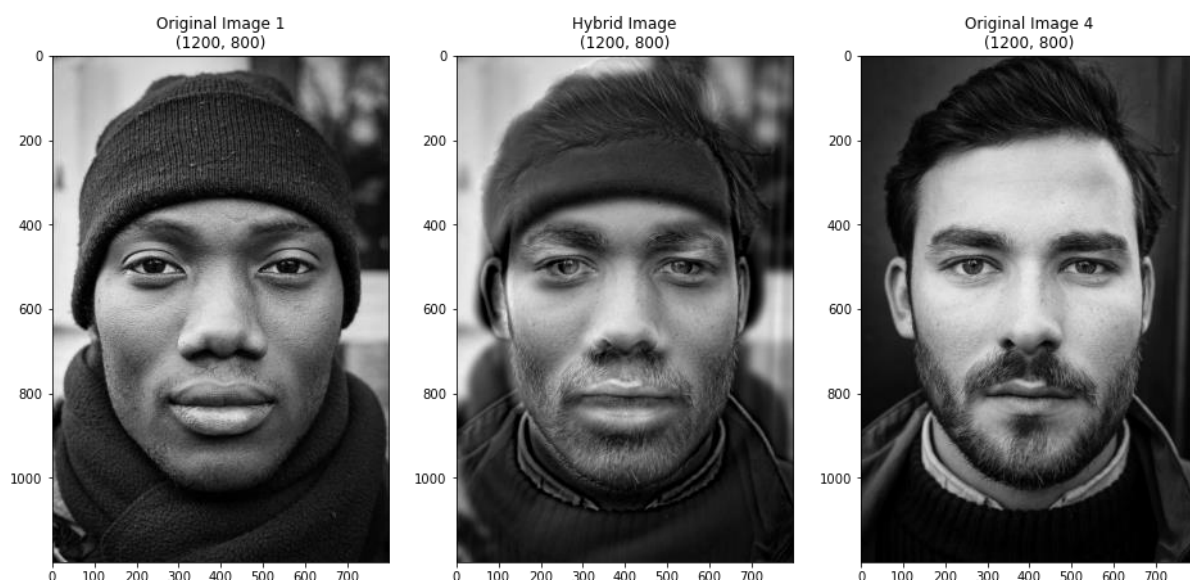
در این قسمت تابع `H4_Hybrid` فراخوانی شده و تصاویر `src_img[3]` و `src_img[4]` با انحراف معیار فیلتر گوسی $m=10$ و $n=10$ به عنوان ورودی به آن پاس داده می شوند. سپس در ادامه تصویر حاصل از ترکیب دو تصویر اولیه و نیز تصاویر ورودی تابع نمایش داده می شوند که مطابق شکل (۳) می باشد. واضح است که اگر از نزدیک به تصویر ترکیبی نگاه کنیم تصویر فرکانس بالا دیده می شود این در حالی است که اگر از دور به تصویر نگاه کنیم تصویر فرکانس پایین بیشتر به چشم می خورد.



شکل (۳)

بلوک [8] فایل ipynb: فراخوانی تابع `H4_Hybrid()` و نمایش تصویر خروجی و تصاویر اولیه

در این قسمت تابع `H4_Hybrid` فراخوانی شده و تصاویر `src_img[4]` و `src_img[1]` با انحراف معیار فیلتر گوسی $m=10$ و $n=10$ به عنوان ورودی به آن پاس داده می شوند. سپس در ادامه تصویر حاصل از ترکیب دو تصویر اولیه و نیز تصاویر اولیه ورودی تابع نمایش داده می شوند که مطابق شکل (۴) می باشد.



شکل (۴)