

pratica

October 25, 2022

```
[8]: !pip install -q nbconvert
!apt-get install -q texlive-xetex texlive-fonts-recommended
↪texlive-plain-generic
```

Reading package lists...

Building dependency tree...

Reading state information...

The following package was automatically installed and is no longer required:

libnvidia-common-460

Use 'apt autoremove' to remove it.

The following additional packages will be installed:

fonts-droid-fallback fonts-lato fonts-lmodern fonts- noto-mono fonts-texgyre
javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
libruby2.5 libsynchronet1 libtexlua52 libtexluajit2 libzip-0-13 lmodern
poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
rubygems-integration t1utils tex-common tex-gyre texlive-base
texlive-binaries texlive-latex-base texlive-latex-extra
texlive-latex-recommended texlive-pictures tipa

Suggested packages:

fonts-noto apache2 | lighttpd | httpd poppler-utils ghostscript
fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic
| fonts-ipafont-gothic fonts-arphic-ukai fonts-arphic-uming fonts-nanum ri
ruby-dev bundler debhelper gv | postscript-viewer perl-tk xpdf-reader
| pdf-viewer texlive-fonts-recommended-doc texlive-latex-base-doc
python-pygments icc-profiles libfile-which-perl
libspreadsheet-parseexcel-perl texlive-latex-extra-doc
texlive-latex-recommended-doc texlive-pstricks dot2tex prerex ruby-tcltk
| libtcltk-ruby texlive-pictures-doc vprerex

The following NEW packages will be installed:

fonts-droid-fallback fonts-lato fonts-lmodern fonts- noto-mono fonts-texgyre
javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
libruby2.5 libsynchronet1 libtexlua52 libtexluajit2 libzip-0-13 lmodern
poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
rubygems-integration t1utils tex-common tex-gyre texlive-base

```

texlive-binaries texlive-fonts-recommended texlive-latex-base
texlive-latex-extra texlive-latex-recommended texlive-pictures
texlive-plain-generic texlive-xetex tipa
0 upgraded, 46 newly installed, 0 to remove and 27 not upgraded.
Need to get 146 MB of archives.
After this operation, 460 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-droid-fallback
all 1:6.0.1r16-1.1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lato all 2.0-2
[2,698 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 poppler-data all
0.4.8-2 [1,479 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 tex-common all 6.09
[33.0 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lmodern all
2.004.5-3 [4,551 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-noto-mono all
20171026-2 [75.5 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe amd64 fonts-texgyre all
20160520-1 [8,761 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 javascript-common all
11 [6,066 B]
Get:9 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsfilters1
amd64 1.20.2-0ubuntu3.1 [108 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsimage2
amd64 2.2.7-1ubuntu2.9 [18.6 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/main amd64 libijs-0.35 amd64
0.35-13 [15.5 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjbig2dec0 amd64
0.13-6 [55.9 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9-common
all 9.26~dfsg+0-0ubuntu0.18.04.17 [5,092 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9 amd64
9.26~dfsg+0-0ubuntu0.18.04.17 [2,267 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjs-jquery all
3.2.1-1 [152 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libkpathsea6
amd64 2017.20170613.44572-8ubuntu0.1 [54.9 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic/main amd64 libpotrace0 amd64
1.14-2 [17.4 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libptexenc1
amd64 2017.20170613.44572-8ubuntu0.1 [34.5 kB]
Get:19 http://archive.ubuntu.com/ubuntu bionic/main amd64 rubygems-integration
all 1.11 [4,994 B]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 ruby2.5 amd64
2.5.1-1ubuntu1.12 [48.6 kB]
Get:21 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby amd64 1:2.5.1
[5,712 B]

```

Get:22 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 rake all
 12.3.1-1ubuntu0.1 [44.9 kB]
 Get:23 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 ruby-did-you-mean all
 1.2.0-2 [9,700 B]
 Get:24 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 ruby-minitest all
 5.10.3-1 [38.6 kB]
 Get:25 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 ruby-net-telnet all
 0.1.1-2 [12.6 kB]
 Get:26 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 ruby-power-assert all
 0.3.0-1 [7,952 B]
 Get:27 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 ruby-test-unit all
 3.2.5-1 [61.1 kB]
 Get:28 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libruby2.5
 amd64 2.5.1-1ubuntu1.12 [3,073 kB]
 Get:29 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libsyntax
 amd64 2017.20170613.44572-8ubuntu0.1 [41.4 kB]
 Get:30 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libtexlua52
 amd64 2017.20170613.44572-8ubuntu0.1 [91.2 kB]
 Get:31 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libtexluajit2
 amd64 2017.20170613.44572-8ubuntu0.1 [230 kB]
 Get:32 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libzip-0-13
 amd64 0.13.62-3.1ubuntu0.18.04.1 [26.0 kB]
 Get:33 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 lmodern all 2.004.5-3
 [9,631 kB]
 Get:34 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 preview-latex-style
 all 11.91-1ubuntu1 [185 kB]
 Get:35 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 t1utils amd64 1.41-2
 [56.0 kB]
 Get:36 <http://archive.ubuntu.com/ubuntu> bionic/universe amd64 tex-gyre all
 20160520-1 [4,998 kB]
 Get:37 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 texlive-
 binaries amd64 2017.20170613.44572-8ubuntu0.1 [8,179 kB]
 Get:38 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 texlive-base all
 2017.20180305-1 [18.7 MB]
 Get:39 <http://archive.ubuntu.com/ubuntu> bionic/universe amd64 texlive-fonts-
 recommended all 2017.20180305-1 [5,262 kB]
 Get:40 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 texlive-latex-base all
 2017.20180305-1 [951 kB]
 Get:41 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 texlive-latex-
 recommended all 2017.20180305-1 [14.9 MB]
 Get:42 <http://archive.ubuntu.com/ubuntu> bionic/universe amd64 texlive-pictures
 all 2017.20180305-1 [4,026 kB]
 Get:43 <http://archive.ubuntu.com/ubuntu> bionic/universe amd64 texlive-latex-
 extra all 2017.20180305-2 [10.6 MB]
 Get:44 <http://archive.ubuntu.com/ubuntu> bionic/universe amd64 texlive-plain-
 generic all 2017.20180305-2 [23.6 MB]
 Get:45 <http://archive.ubuntu.com/ubuntu> bionic/universe amd64 tipa all 2:1.3-20
 [2,978 kB]

```

Get:46 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-xetex all
2017.20180305-1 [10.7 MB]
Fetched 146 MB in 12s (12.5 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages ...
Selecting previously unselected package fonts-droid-fallback.
(Reading database ... 123942 files and directories currently installed.)
Preparing to unpack .../00-fonts-droid-fallback_1%3a6.0.1r16-1.1_all.deb ...
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1) ...
Selecting previously unselected package fonts-lato.
Preparing to unpack .../01-fonts-lato_2.0-2_all.deb ...
Unpacking fonts-lato (2.0-2) ...
Selecting previously unselected package poppler-data.
Preparing to unpack .../02-poppler-data_0.4.8-2_all.deb ...
Unpacking poppler-data (0.4.8-2) ...
Selecting previously unselected package tex-common.
Preparing to unpack .../03-tex-common_6.09_all.deb ...
Unpacking tex-common (6.09) ...
Selecting previously unselected package fonts-lmodern.
Preparing to unpack .../04-fonts-lmodern_2.004.5-3_all.deb ...
Unpacking fonts-lmodern (2.004.5-3) ...
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack .../05-fonts-noto-mono_20171026-2_all.deb ...
Unpacking fonts-noto-mono (20171026-2) ...
Selecting previously unselected package fonts-texgyre.
Preparing to unpack .../06-fonts-texgyre_20160520-1_all.deb ...
Unpacking fonts-texgyre (20160520-1) ...
Selecting previously unselected package javascript-common.
Preparing to unpack .../07-javascript-common_11_all.deb ...
Unpacking javascript-common (11) ...
Selecting previously unselected package libcupsfilters1:amd64.
Preparing to unpack .../08-libcupsfilters1_1.20.2-0ubuntu3.1_amd64.deb ...
Unpacking libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) ...
Selecting previously unselected package libcupsimage2:amd64.
Preparing to unpack .../09-libcupsimage2_2.2.7-1ubuntu2.9_amd64.deb ...
Unpacking libcupsimage2:amd64 (2.2.7-1ubuntu2.9) ...
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack .../10-libijs-0.35_0.35-13_amd64.deb ...
Unpacking libijs-0.35:amd64 (0.35-13) ...
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack .../11-libjbig2dec0_0.13-6_amd64.deb ...
Unpacking libjbig2dec0:amd64 (0.13-6) ...
Selecting previously unselected package libgs9-common.
Preparing to unpack .../12-libgs9-common_9.26~dfsg+0-0ubuntu0.18.04.17_all.deb
...
Unpacking libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.17) ...
Selecting previously unselected package libgs9:amd64.
Preparing to unpack .../13-libgs9_9.26~dfsg+0-0ubuntu0.18.04.17_amd64.deb ...

```

```

Unpacking libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.17) ...
Selecting previously unselected package libjs-jquery.
Preparing to unpack .../14-libjs-jquery_3.2.1-1_all.deb ...
Unpacking libjs-jquery (3.2.1-1) ...
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack .../15-libkpathsea6_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libpotrace0.
Preparing to unpack .../16-libpotrace0_1.14-2_amd64.deb ...
Unpacking libpotrace0 (1.14-2) ...
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack .../17-libptexenc1_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package rubygems-integration.
Preparing to unpack .../18-rubygems-integration_1.11_all.deb ...
Unpacking rubygems-integration (1.11) ...
Selecting previously unselected package ruby2.5.
Preparing to unpack .../19-ruby2.5_2.5.1-1ubuntu1.12_amd64.deb ...
Unpacking ruby2.5 (2.5.1-1ubuntu1.12) ...
Selecting previously unselected package ruby.
Preparing to unpack .../20-ruby_1%3a2.5.1_amd64.deb ...
Unpacking ruby (1:2.5.1) ...
Selecting previously unselected package rake.
Preparing to unpack .../21-rake_12.3.1-1ubuntu0.1_all.deb ...
Unpacking rake (12.3.1-1ubuntu0.1) ...
Selecting previously unselected package ruby-did-you-mean.
Preparing to unpack .../22-ruby-did-you-mean_1.2.0-2_all.deb ...
Unpacking ruby-did-you-mean (1.2.0-2) ...
Selecting previously unselected package ruby-minitest.
Preparing to unpack .../23-ruby-minitest_5.10.3-1_all.deb ...
Unpacking ruby-minitest (5.10.3-1) ...
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack .../24-ruby-net-telnet_0.1.1-2_all.deb ...
Unpacking ruby-net-telnet (0.1.1-2) ...
Selecting previously unselected package ruby-power-assert.
Preparing to unpack .../25-ruby-power-assert_0.3.0-1_all.deb ...
Unpacking ruby-power-assert (0.3.0-1) ...
Selecting previously unselected package ruby-test-unit.
Preparing to unpack .../26-ruby-test-unit_3.2.5-1_all.deb ...
Unpacking ruby-test-unit (3.2.5-1) ...
Selecting previously unselected package libruby2.5:amd64.
Preparing to unpack .../27-libruby2.5_2.5.1-1ubuntu1.12_amd64.deb ...
Unpacking libruby2.5:amd64 (2.5.1-1ubuntu1.12) ...
Selecting previously unselected package libsyntax1:amd64.
Preparing to unpack .../28-libsyntax1_2017.20170613.44572-8ubuntu0.1_amd64.deb
...

```

```

Unpacking libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libtexlua52:amd64.
Preparing to unpack .../29-libtexlua52_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libtexluaajit2:amd64.
Preparing to unpack
.../30-libtexluaajit2_2017.20170613.44572-8ubuntu0.1_amd64.deb ...
Unpacking libtexluaajit2:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libbzip-0-13:amd64.
Preparing to unpack .../31-libbzip-0-13_0.13.62-3.1ubuntu0.18.04.1_amd64.deb ...
Unpacking libbzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) ...
Selecting previously unselected package lmodern.
Preparing to unpack .../32-lmodern_2.004.5-3_all.deb ...
Unpacking lmodern (2.004.5-3) ...
Selecting previously unselected package preview-latex-style.
Preparing to unpack .../33-preview-latex-style_11.91-1ubuntu1_all.deb ...
Unpacking preview-latex-style (11.91-1ubuntu1) ...
Selecting previously unselected package t1utils.
Preparing to unpack .../34-t1utils_1.41-2_amd64.deb ...
Unpacking t1utils (1.41-2) ...
Selecting previously unselected package tex-gyre.
Preparing to unpack .../35-tex-gyre_20160520-1_all.deb ...
Unpacking tex-gyre (20160520-1) ...
Selecting previously unselected package texlive-binaries.
Preparing to unpack .../36-texlive-
binaries_2017.20170613.44572-8ubuntu0.1_amd64.deb ...
Unpacking texlive-binaries (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package texlive-base.
Preparing to unpack .../37-texlive-base_2017.20180305-1_all.deb ...
Unpacking texlive-base (2017.20180305-1) ...
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack .../38-texlive-fonts-recommended_2017.20180305-1_all.deb ...
Unpacking texlive-fonts-recommended (2017.20180305-1) ...
Selecting previously unselected package texlive-latex-base.
Preparing to unpack .../39-texlive-latex-base_2017.20180305-1_all.deb ...
Unpacking texlive-latex-base (2017.20180305-1) ...
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack .../40-texlive-latex-recommended_2017.20180305-1_all.deb ...
Unpacking texlive-latex-recommended (2017.20180305-1) ...
Selecting previously unselected package texlive-pictures.
Preparing to unpack .../41-texlive-pictures_2017.20180305-1_all.deb ...
Unpacking texlive-pictures (2017.20180305-1) ...
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack .../42-texlive-latex-extra_2017.20180305-2_all.deb ...
Unpacking texlive-latex-extra (2017.20180305-2) ...
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack .../43-texlive-plain-generic_2017.20180305-2_all.deb ...

```

```

Unpacking texlive-plain-generic (2017.20180305-2) ...
Selecting previously unselected package tipa.
Preparing to unpack .../44-tipa_2%3a1.3-20_all.deb ...
Unpacking tipa (2:1.3-20) ...
Selecting previously unselected package texlive-xetex.
Preparing to unpack .../45-texlive-xetex_2017.20180305-1_all.deb ...
Unpacking texlive-xetex (2017.20180305-1) ...
Setting up libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.17) ...
Setting up libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Setting up libjs-jquery (3.2.1-1) ...
Setting up libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Setting up fonts-droid-fallback (1:6.0.1r16-1.1) ...
Setting up libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Setting up libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Setting up tex-common (6.09) ...
update-language: texlive-base not installed and configured, doing nothing!
Setting up poppler-data (0.4.8-2) ...
Setting up tex-gyre (20160520-1) ...
Setting up preview-latex-style (11.91-1ubuntu1) ...
Setting up fonts-texgyre (20160520-1) ...
Setting up fonts-noto-mono (20171026-2) ...
Setting up fonts-lato (2.0-2) ...
Setting up libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) ...
Setting up libcupsimage2:amd64 (2.2.7-1ubuntu2.9) ...
Setting up libjbig2dec0:amd64 (0.13-6) ...
Setting up ruby-did-you-mean (1.2.0-2) ...
Setting up t1utils (1.41-2) ...
Setting up ruby-net-telnet (0.1.1-2) ...
Setting up libijs-0.35:amd64 (0.35-13) ...
Setting up rubygems-integration (1.11) ...
Setting up libpotrace0 (1.14-2) ...
Setting up javascript-common (11) ...
Setting up ruby-minitest (5.10.3-1) ...
Setting up libzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) ...
Setting up libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.17) ...
Setting up libtexluaajit2:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Setting up fonts-lmodern (2.004.5-3) ...
Setting up ruby-power-assert (0.3.0-1) ...
Setting up texlive-binaries (2017.20170613.44572-8ubuntu0.1) ...
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up texlive-base (2017.20180305-1) ...
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST...
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN...
mktexlsr: Updating /var/lib/texmf/ls-R...
mktexlsr: Done.

```

```

tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4:
/var/lib/texmf/tex/generic/config/pdftexconfig.tex
Setting up texlive-fonts-recommended (2017.20180305-1) ...
Setting up texlive-plain-generic (2017.20180305-2) ...
Setting up texlive-latex-base (2017.20180305-1) ...
Setting up lmodern (2.004.5-3) ...
Setting up texlive-latex-recommended (2017.20180305-1) ...
Setting up texlive-pictures (2017.20180305-1) ...
Setting up tipa (2:1.3-20) ...
Regenerating '/var/lib/texmf/fmtutil.cnf-DEBIAN'... done.
Regenerating '/var/lib/texmf/fmtutil.cnf-TEXLIVEDIST'... done.
update-fmtutil has updated the following file(s):
    /var/lib/texmf/fmtutil.cnf-DEBIAN
    /var/lib/texmf/fmtutil.cnf-TEXLIVEDIST
If you want to activate the changes in the above file(s),
you should run fmtutil-sys or fmtutil.
Setting up texlive-latex-extra (2017.20180305-2) ...
Setting up texlive-xetex (2017.20180305-1) ...
Setting up ruby2.5 (2.5.1-1ubuntu1.12) ...
Setting up ruby (1:2.5.1) ...
Setting up ruby-test-unit (3.2.5-1) ...
Setting up rake (12.3.1-1ubuntu0.1) ...
Setting up libruby2.5:amd64 (2.5.1-1ubuntu1.12) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.6) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for fontconfig (2.12.6-0ubuntu2) ...
Processing triggers for tex-common (6.09) ...
Running updmap-sys. This may take some time... done.
Running mktexlsr /var/lib/texmf ... done.
Building format(s) --all.
    This may take some time... done.

```

1 Introdução

Grupo: Germano Barcelos (3873), Guilherme Melos (3882), Jhonata Miranda (3859)

Neste relatório, serão introduzidos e explorados em prática os conceitos de espaço de cores e segmentação de imagens utilizando as bibliotecas *OpenCV*, *Numpy*, *Matplotlib* e *Scipy* para realizar as atividades propostas.

Neste relatório apresentaremos alguns espaços de cores vistos em aula. Os espaços trabalhados

aqui são: BGR, HSV e LAB. O espaço de cores BGR, também conhecido como RGB, é definido pelas três cromaticidades das primárias aditivas vermelha, verde e azul e pode produzir qualquer cromaticidade que seja o triângulo definido por essas cores primárias. A especificação completa de um espaço de cores RGB também requer uma cromaticidade de ponto branco e uma curva de correção de gama.

Espaços de cor de matiz, saturação e valor (HSV) são frequentemente usados por artistas. "Hue" é o que normalmente pensamos como cor. É o atributo de uma cor pela qual damos a ela um nome como "vermelho" ou "azul". "Valor" é outra palavra para "leveza", o atributo de uma cor que faz parecer equivalente a algum tom de cinza entre preto e branco. A saturação é uma medida de quão diferente uma cor aparece de um cinza da mesma leveza. A saturação zero indica nenhuma tonalidade, apenas a escala cinza. O espaço de cor HSV é normalizado.

O espaço de cor LAB, também conhecido como espaço de cor CIELAB é atualmente o mais popular dos espaços de cores uniformes usados para avaliar as cores. Esse espaço de cor é amplamente utilizado pois correlaciona consistentemente os valores de cor com a percepção visual. Indústrias como as de plástico, tintas, impressão, alimentos e têxtil, além de universidades, utilizam este espaço para identificar, comunicar e avaliar os atributos da cor além das inconsistências ou desvios de uma cor padrão.

2 Desenvolvimento

Neste relatório, realizamos os seguintes exercícios propostos em aula:

1. Apresentar o script e discutir o script 2 da Aula 2
2. Usar o script 2 para apresentar imagem em LAB, HSV, RGB, mostrando uma imagem de uma flor, de rostos, de umas 4 pessoas e de uma praia, mostrando o mar. Você pode fazer uma montagem, usando o PAINT mesmo, para criar uma imagem contendo as 3 cenas acima, e apresentar nos 3 espaços de cor. Para cada espaço de cor, trace as 3 bandas isoladas e analise a variação dos objetos na mesma cena.
3. Com a imagem cubos.png, crie mais duas imagens, uma mais escura e outra mais clara. Converta as 3 imagens para HSV e as bandas separadas, e analise como a imagem varia em cada banda, ou seja, quais as diferenças na banda H, S e V para as 3 imagens (escura, normal e clara)
4. Usando o filme de gols cruzeiro x inter, aplique um filtro de média para uniformizar a imagem, obtenha duas máscaras binárias, sendo uma uma segmentação utilizando a faixa de valores em RGB, e outra, em HSV (veja qual melhor resultado para criar a máscara com espaço HSV - só H, S ou V, ou uma combinação entre elas), aplique erosão/dilatação destas duas máscaras para tentar obter dois filmes: Um obtido da segmentação utilizando a faixa de valores em RGB, e outra, em HSV. Pode dilatar muito a máscara para mostrar mais do jogador. O filme resultante deve ser um AND entre a máscara obtida e o filme original, e com saída em rgb mesmo... passos: filtro mediana, segmentação com valores RGB e outra com valores HSV; nas máscaras resultantes, aplicar dilatação e erosão. Depois, aplicar as duas máscaras no filme, obtendo dois filmes RGB, um a partir da máscara obtida pelos valores rgb e outro pela HSV

2.1 Exercício 1

Primeiramente importamos as bibliotecas OpenCV, Numpy, Matplotlib para executar suas funções:

```
[3]: import cv2 #import openCV2 package
import numpy as np #import numpy as short np
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
```

Após isso, carregamos a imagem `TEXTOS.png`, fazemos a conversão para o espaço de cores HSV, salvamos cada banda em uma variável, definimos também a fonte `HERSHEY_SIMPLEX`, a posição `org`, a escala da fonte como 1, e salvamos a cor azul em BGR. Também definimos a espessura da linha como 2 pixels.

```
[7]: input = cv2.imread("/content/TEXTOS.png")

hsv_image=cv2.cvtColor(input,cv2.COLOR_BGR2HSV)

B,G,R=cv2.split(input)

# font
font = cv2.FONT_HERSHEY_SIMPLEX

# POSICAO
org = (20, 50)

# fontScale
fontScale = 1

# Blue color in BGR
color = (255, 255, 255)

# Line thickness of 2 px
thickness = 2
```

Após isso, usamos a função `cv2.putText` para que desenha uma string com os parâmetros recebidos. Passamos os parâmetros definidos anteriormente e conseguimos observar o texto colorido e um clareamento da cor de cada faixa a medida em que esta cor vai se aproximando da cor pura. Nota-se que na imagem que utiliza as cores puras, não é possível visualizar a cor correspondente. Já na imagem que mostra os valores BGR com colorações próximas, mas não iguais a original, podemos ainda visualizar a cor e ver que a mesma está se aproximando do branco, enquanto as outras foram clareadas apenas uma certa quantidade, quantidade essa que representa o percentual da cor clareada que está misturada a essa cor.

```
[9]: # Using cv2.putText() method
image = cv2.putText(R, 'R', org, font,
                    fontScale, (0,0,255), thickness, cv2.LINE_AA)
```




```

image = cv2.putText(G, 'G', org, font,
                    fontStyle, (0,255,0), thickness, cv2.LINE_AA)

image = cv2.putText(B, 'B', org, font,
                    fontStyle, (0,255,0), thickness, cv2.LINE_AA)

cv2_imshow(input)
imagem=np.vstack((B,G,R))
cv2_imshow(imagem)

```

	
VERMELHO	234,21,64
	
VERDE	64,213,42
	
AZUL	68,36,220

B		234,21,64
		64,213,42
	VERDE	68,36,220

G		234,21,64
		64,213,42
	AZUL	68,36,220

R		234,21,64
		64,213,42
	AZUL	68,36,220

Abaixo, é feito o mesmo procedimento realizado acima para o espaço de cores BGR mas, com o espaço HSV. Primeiramente, são exibidas as imagens: `MATIZ.png`, `saturação.png` e `brilho.png`. Após isso, exibimos as bandas HSV de cada uma das imagens utilizadas no exemplo acima.

```
[15]: h=cv2.imread("/content/MATIZ.png")
      s=cv2.imread("/content/saturação.png")
      v=cv2.imread("/content/brilho.png")
      plt.figure(figsize = ((12, 6)))
      plt.subplot(1, 3, 1)
      plt.title("Matiz", fontsize = 16)
      plt.imshow(h)
```

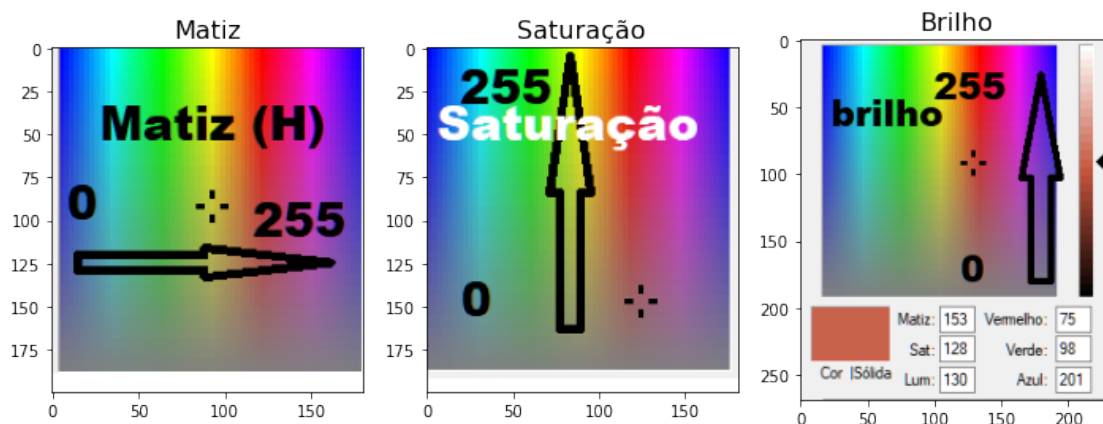
```

plt.subplot(1, 3, 2)
plt.title("Saturação", fontsize = 16)
#plt.suptitle("Gráficos", fontsize = 20)
plt.imshow(s)
plt.subplot(1, 3, 3)
plt.title("Brilho", fontsize = 16)
plt.imshow(v)
plt.show()

H,S,V=cv2.split(hsv_image)
# font
font = cv2.FONT_HERSHEY_SIMPLEX
# POSICAO
org = (20, 50)
# fontScale
fontScale = 1
# Blue color in BGR
color = (255, 255, 255)
# Line thickness of 2 px
thickness = 2
# Using cv2.putText() method
image = cv2.putText(H, 'H', org, font,
                    fontScale, color, thickness, cv2.LINE_AA)
image = cv2.putText(S, 'S', org, font,
                    fontScale, color, thickness, cv2.LINE_AA)
image = cv2.putText(V, 'V', org, font,
                    fontScale, (0,0,0), thickness, cv2.LINE_AA)

cv2_imshow(input)
cv2_imshow(hsv_image)
imagem=np.vstack((H,S,V))
cv2_imshow(imagem)

```



VERMELHO	234,21,64
VERDE	64,213,42
AZUL	68,36,220

VERMELHO	234,21,64
VERDE	64,213,42
AZUL	68,36,220



Abaixo, trabalhamos com a banda V da imagem original, gerando o histograma e equalizando o mesmo. Após isso é exibido a banda V antes e depois da equalização e o seus respectivos histogramas. Por fim, mostramos a imagem com HSV equalizado e com BGR equalizado a partir do BGR:

```
[20]: print(np.max(np.max(V)))

#imagem original V
V=np.uint8(V)
#Histograma de V
V_hist = cv2.calcHist([V],[0],None,[256],[0,256])
```

```

# Equalizando V
V_equ = cv2.equalizeHist(V)
V_equ=np.uint8(V_equ)
# Histograma equalizado
V_hist_equ = cv2.calcHist([V_equ],[0],None,[256],[0,256])

plt.figure(figsize = ((12, 6)))
plt.subplot(2, 2, 1)
plt.title("Banda V", fontsize = 16)
plt.imshow(V)

plt.subplot(2, 2, 2)
plt.title("Banda V Equalizada", fontsize = 16)
#plt.suptitle("Gráficos", fontsize = 20)
plt.imshow(V_equ)

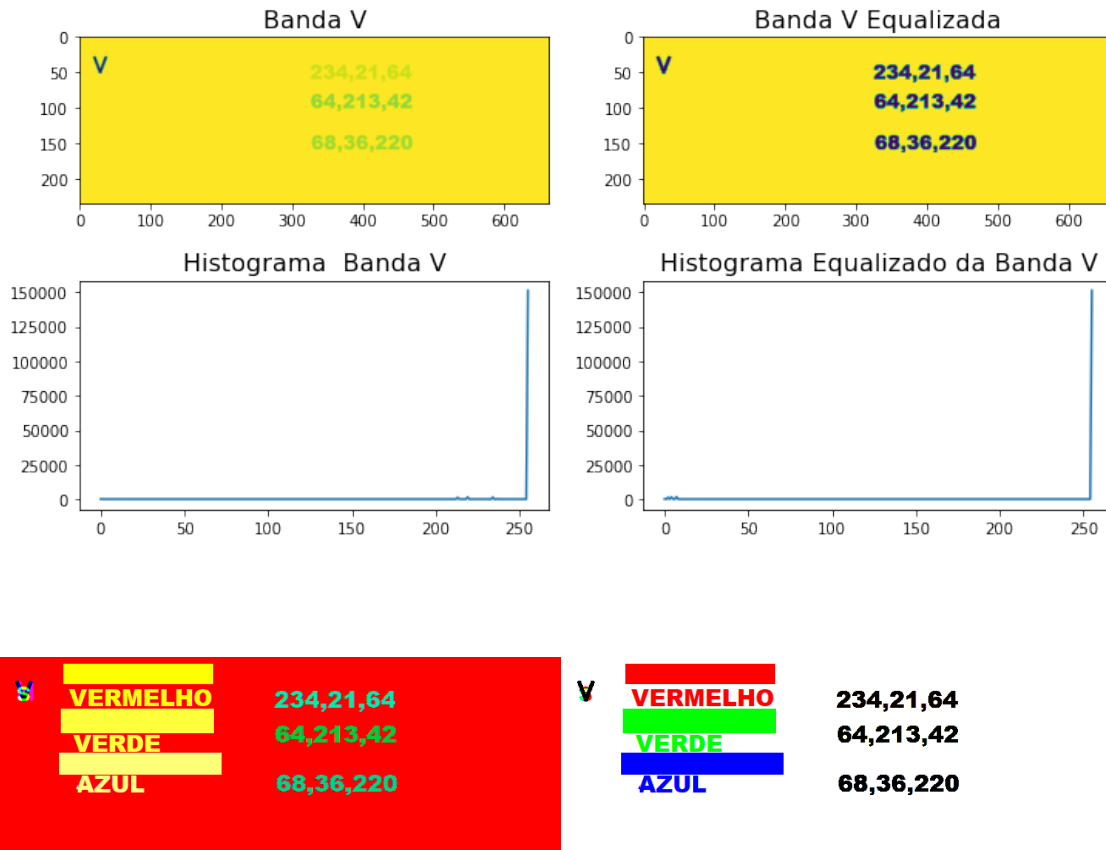
plt.subplot(2, 2, 3)
plt.title("Histograma Banda V", fontsize = 16)
plt.suptitle("Gráficos", fontsize = 20)
plt.plot(V_hist)

plt.subplot(2, 2, 4)
plt.title("Histograma Equalizado da Banda V", fontsize = 16)
plt.suptitle("Gráficos", fontsize = 20)
plt.plot(V_hist_equ)
plt.show()

#mostrando hsv equalizado e rgb equalizado a partir do hsv
hsv=cv2.merge([H,S,V_equ])
rgb=cv2.cvtColor(hsv,cv2.COLOR_HSV2BGR)
imagem=np.hstack((hsv,rgb))
cv2_imshow(imagem)

```


Gráficos



2.2 Exercício 2

Para esse exercício, importamos as bibliotecas OpenCV, Numpy, Matplotlib, ArgParse, Glob e Time.

```
[27]: import cv2
import time
import argparse
import glob
import numpy as np
import matplotlib.pyplot as plt
```

Aqui, definimos duas funções: a função `plot_images` é utilizada para plotar as imagens, já configuradas, utilizando a biblioteca Matplotlib, já a função `get_spacecolor` é utilizada para converter uma imagem original para os espaços de cores HSV, BGR e LAB.

```
[22]: def plot_images(imgs, titles, x, y, figsize, cmap="viridis"):
images_list_w_titles = list(zip(imgs, titles))
```

```

f, axarr = plt.subplots(x,y, figsize=figsize)
for i in range(x):
    for j in range(y):
        if x > 1:
            axarr[i, j].imshow(np.uint8(images_list_w_titles[(i*y)+j][0]),  

→ cmap=cmap)
            axarr[i, j].set_title(images_list_w_titles[(i*y)+j][1])
        else:
            axarr[j].imshow(np.uint8(images_list_w_titles[(i*y)+j][0]),  

→ cmap=cmap)
            axarr[j].set_title(images_list_w_titles[(i*y)+j][1])

def get_spacecolor(original):
    scale_percent = 60 # percent of original size
    width = int(original.shape[1] * scale_percent / 100)
    height = int(original.shape[0] * scale_percent / 100)
    dim = (width, height)

    original = cv2.resize(original, dim, interpolation = cv2.INTER_AREA)

    # Convert the BGR image to other color spaces
    imageBGR = cv2.cvtColor(original,cv2.COLOR_BGR2RGB)
    imageHSV = cv2.cvtColor(original,cv2.COLOR_BGR2HSV)
    imageLAB = cv2.cvtColor(original,cv2.COLOR_BGR2LAB)
    return imageBGR, imageHSV, imageLAB

```

2.2.1 Rostos

Utilizamos a imagem `colagem rostos.jpg` como referência para exibi-las nos diferentes espaços de cores utilizando as funções definidas acima. Como resultado, plotamos as imagens originais, as conversões e cada uma das faixas de cor de cada um dos espaços.

```

[23]: original=cv2.imread("/content/colagem rostos.jpg")

imageBGR, imageHSV, imageLAB = get_spacecolor(original)

imgs = [imageBGR, imageHSV, imageLAB]
titles = ["RBG", "HSV", "LAB"]
plot_images(imgs, titles, 1, 3, (20, 20))

R,G,B = cv2.split(imageBGR)
H,S,V = cv2.split(imageHSV)
L,A,B = cv2.split(imageLAB)

imgs = [R, G, B, H, S, V, L, A, B]

```

```
titles = ["R", "G", "B", "H", "S", "V", "L", "A", "B"]
plot_images(imgs, titles, 3, 3, (20, 15), cmap="gray")
```



2.2.2 Flor

Por conseguinte, realizamos o mesmo procedimento acima para a imagem `flor.jpg`:

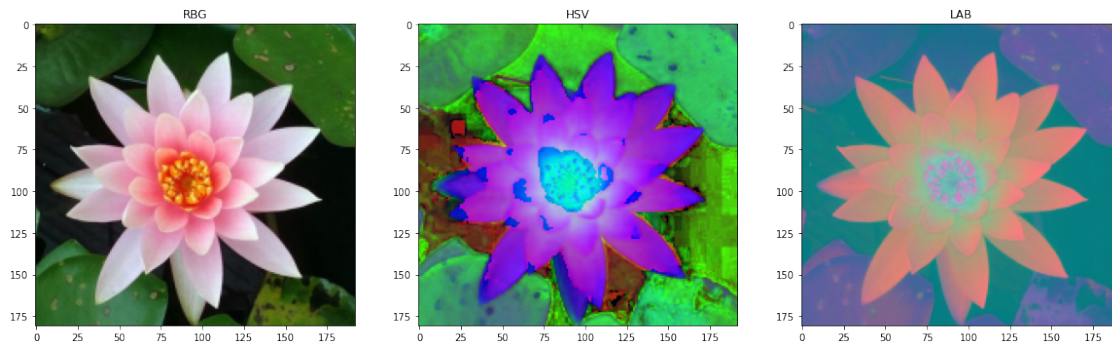
```
[24]: original=cv2.imread("/content/flor.jpg")

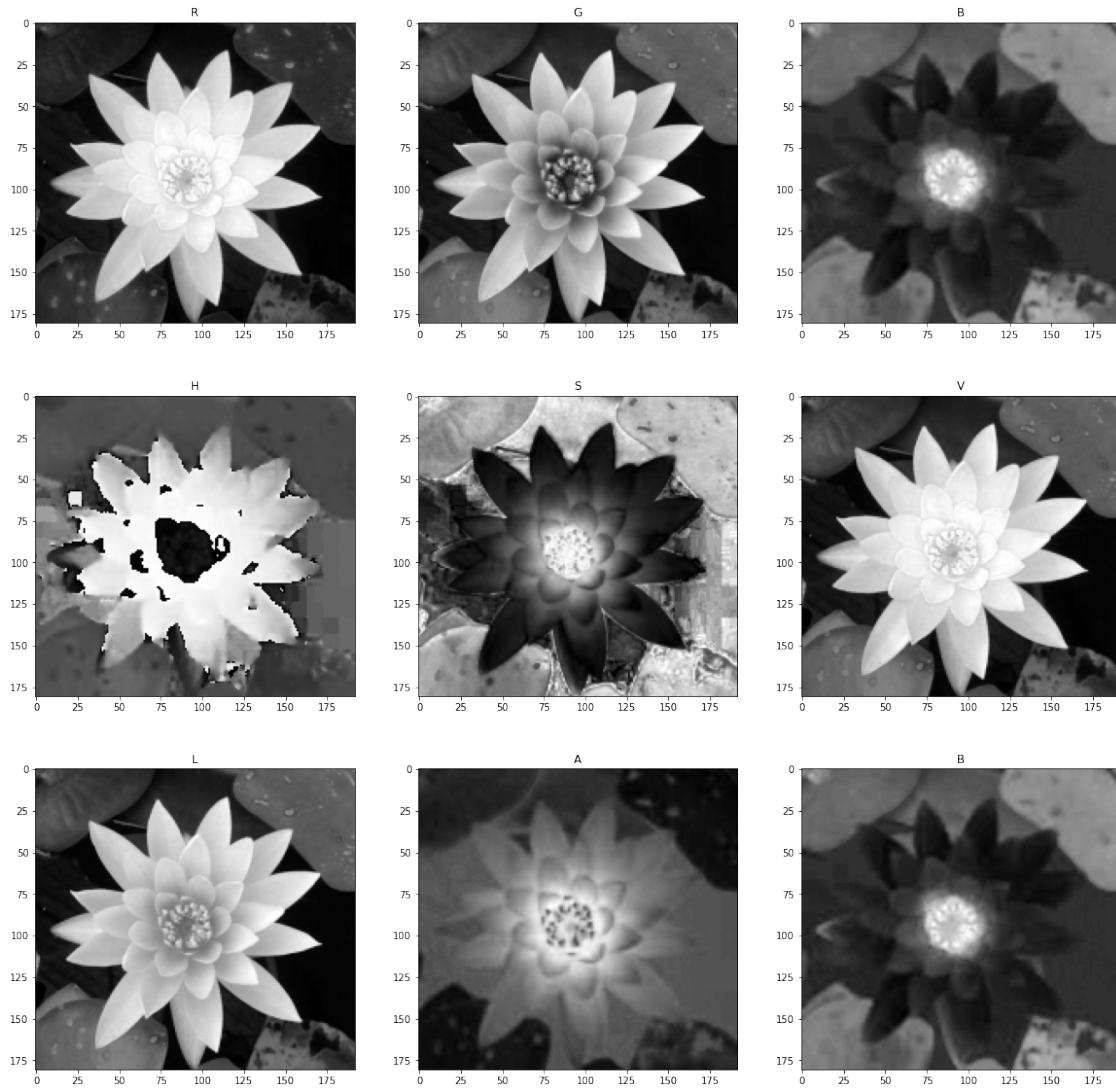
imageBGR, imageHSV, imageLAB = get_spacecolor(original)

imgs = [imageBGR, imageHSV, imageLAB]
titles = ["RBG", "HSV", "LAB"]
plot_images(imgs, titles, 1, 3, (20, 20))

R,G,B = cv2.split(imageBGR)
H,S,V = cv2.split(imageHSV)
L,A,B = cv2.split(imageLAB)

imgs = [R, G, B, H, S, V, L, A, B]
titles = ["R", "G", "B", "H", "S", "V", "L", "A", "B"]
plot_images(imgs, titles, 3, 3, (20, 20), cmap="gray")
```





2.2.3 Praia

Fizemos também o mesmo procedimento para a imagem `praia.jpg`:

```
[26]: original=cv2.imread("/content/praia.jpg")

imageBGR, imageHSV, imageLAB = get_spacecolor(original)

imgs = [imageBGR, imageHSV, imageLAB]
titles = ["RGB", "HSV", "LAB"]
plot_images(imgs, titles, 1, 3, (20, 20))

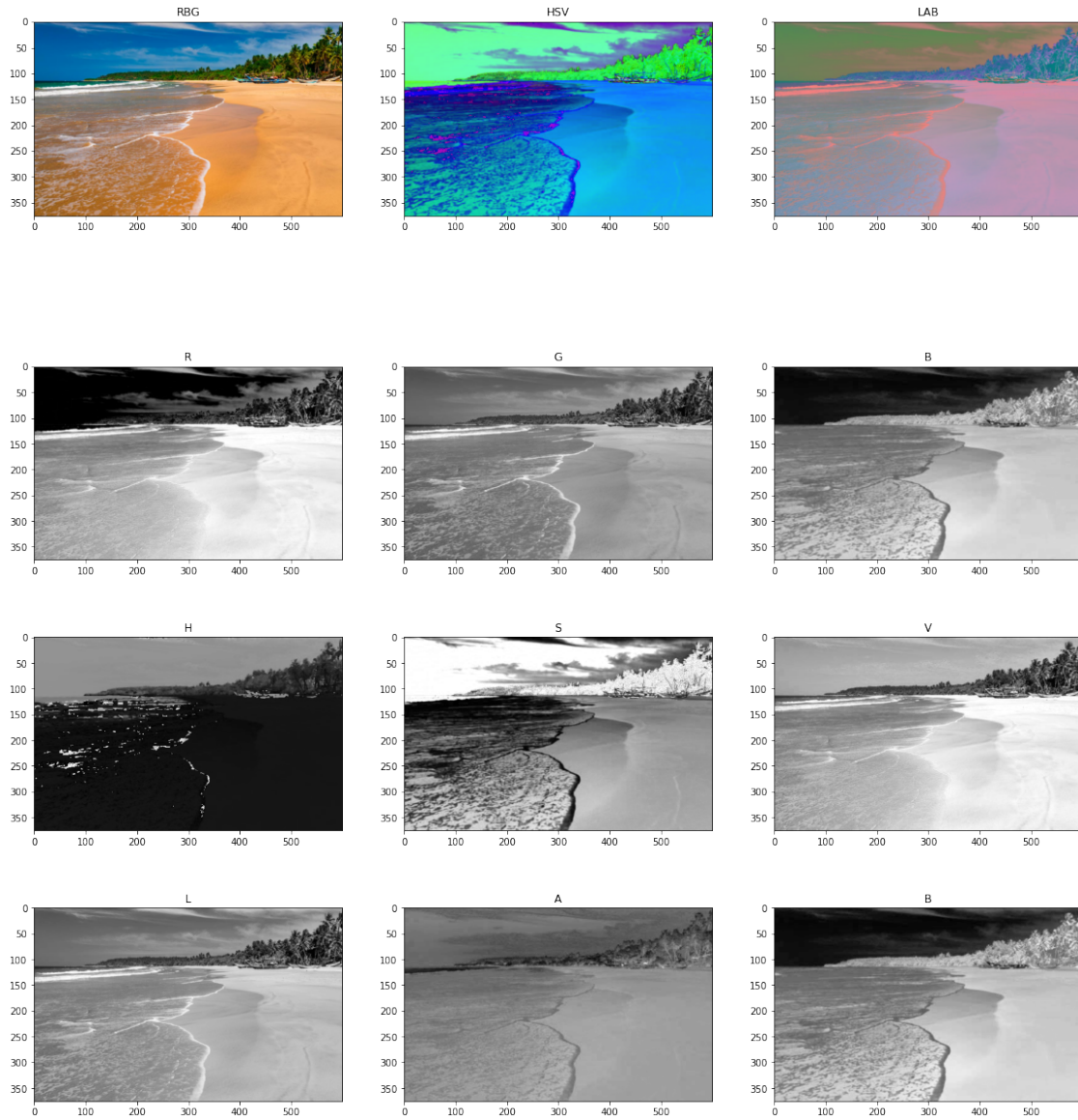
R,G,B = cv2.split(imageBGR)
```

```

H,S,V = cv2.split(imageHSV)
L,A,B = cv2.split(imageLAB)

imgs = [R, G, B, H, S, V, L, A, B]
titles = ["R", "G", "B", "H", "S", "V", "L", "A", "B"]
plot_images(imgs, titles, 3, 3, (20, 15), cmap="gray")

```



Por fim, utilizamos imagens com diferente tonalidades de cores, brilhos diferente e tentamos diversificar o tipo de imagem para que possamos distinguir os espaços de cores e visualizar a diferença entre eles.

2.3 Exercício 3

Para o exercício abaixo, utilizamos as bibliotecas OpenCv e Numpy.

```
[5]: import cv2
import numpy as np
from google.colab.patches import cv2_imshow
```

Primeiro reutilizamos a função `controller` para fazer modificações no brilho e contraste de uma imagem e obtermos efeitos mais escuros ou claros dependendo do parâmetro que será passado.

```
[3]: def controller(img, Brilho=255, Contraste=127):
    Brilho = int((Brilho - 0) * (255 - (-255)) / (510 - 0) + (-255))
    Contraste = int((Contraste - 0) * (127 - (-127)) / (254 - 0) + (-127))

    if Brilho != 0:
        if Brilho > 0:
            shadow = Brilho
            max = 255
        else:
            shadow = 0
            max = 255 + Brilho
        al_pha = (max - shadow) / 255
        ga_mma = shadow

        # The function addWeighted calculates the weighted sum of two arrays
        cal = cv2.addWeighted(img, al_pha, img, 0, ga_mma)

    else:
        cal = img

    if Contraste != 0:
        Alpha = float(131 * (Contraste + 127)) / (127 * (131 - Contraste))
        Gamma = 127 * (1 - Alpha)

        cal = cv2.addWeighted(cal, Alpha, cal, 0, Gamma)

    return cal
```

Com a imagem `cubos.png` carregada, aplicamos efeitos para clarear e escurecer a imagem utilizando os parâmetros 150 para brilho e 150 para contraste para escurecer um pouco a imagem e 330 e 170 para clarear um pouco a mesma. Após isso concatenamos a imagem original com as duas resultantes dos efeitos aplicados e exibimos o conjunto. Após converter as 3 imagens para o espaço de cores HSV separamos cada uma das bandas H, S e V de cada imagem, concatenamos as 3 bandas para cada imagem e exibimos cada um dos resultados.

```
[7]: # The function imread loads an
# image from the specified file and returns it.
img_original = cv2.imread("/content/cubos.png")

# Making another copy of an image.
img = img_original.copy()

#aplicando efeitos
img_escura = controller(img,150,150)
img_clara = controller(img,330,170)

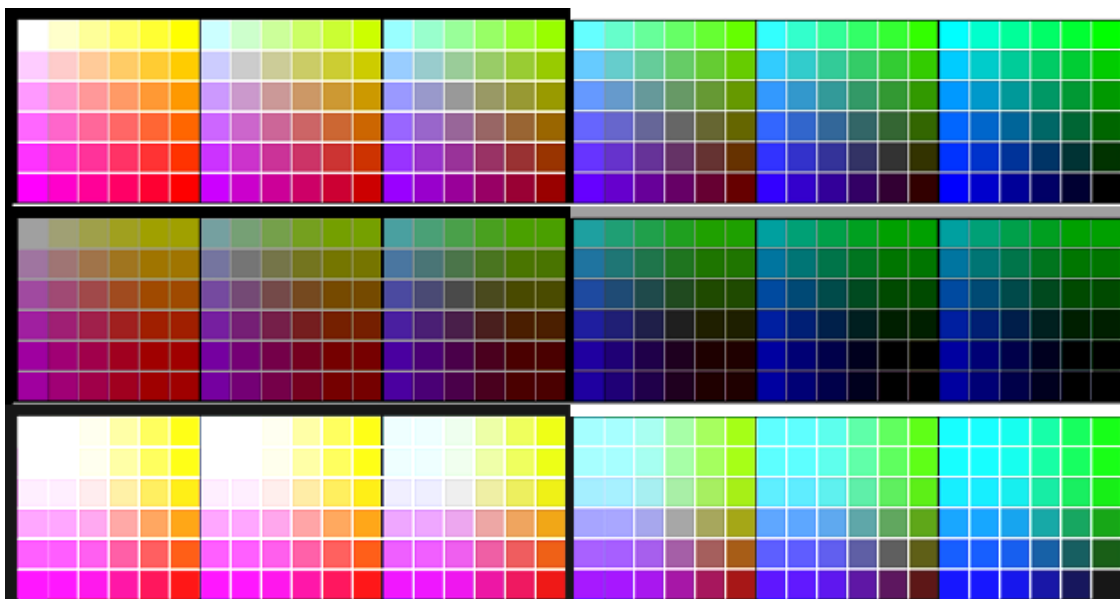
#concatenando e exibindo imagens
img_bgr_concat = cv2.vconcat([img_original,img_escura,img_clara])
print("Imagens: Original + Escura + Clara")
cv2_imshow(img_bgr_concat)

#convertendo para hsv
img_original_hsv = cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
img_escura_hsv = cv2.cvtColor(img_escura,cv2.COLOR_BGR2HSV)
img_clara_hsv = cv2.cvtColor(img_clara,cv2.COLOR_BGR2HSV)

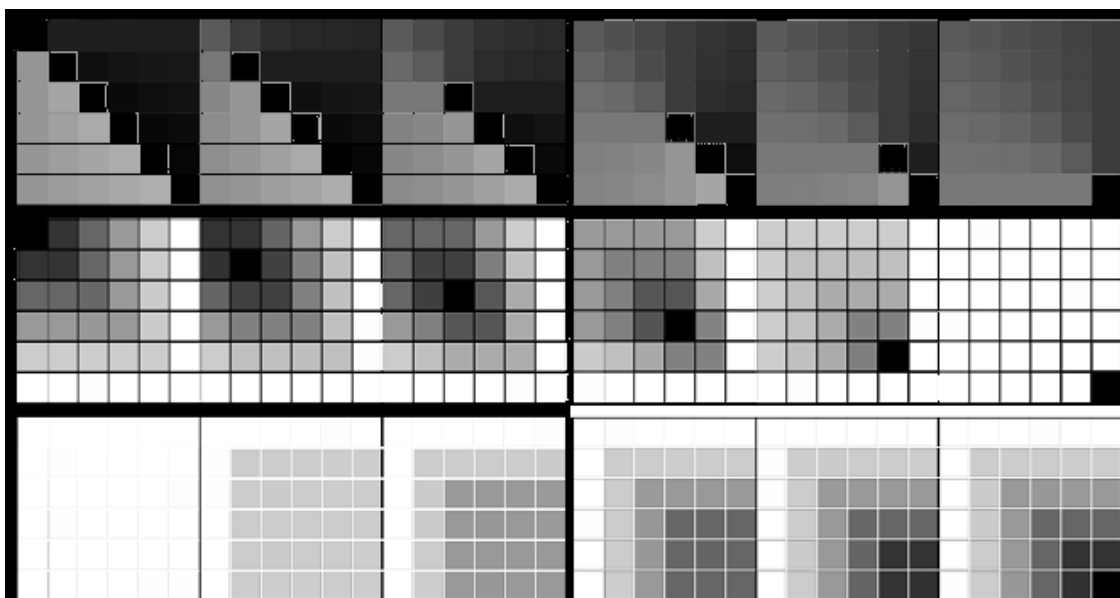
#separando as bandas
img_original_hsv_h, img_original_hsv_s, img_original_hsv_v = cv2.
↪split(img_original_hsv)
img_escura_hsv_h, img_escura_hsv_s, img_escura_hsv_v = cv2.
↪split(img_escura_hsv)
img_clara_hsv_h, img_clara_hsv_s, img_clara_hsv_v = cv2.split(img_clara_hsv)

#concatenando e exibindo
img_original_hsv_concat = cv2.
↪vconcat([img_original_hsv_h,img_original_hsv_s,img_original_hsv_v])
img_escura_hsv_concat = cv2.
↪vconcat([img_escura_hsv_h,img_escura_hsv_s,img_escura_hsv_v])
img_clara_hsv_concat = cv2.
↪vconcat([img_clara_hsv_h,img_clara_hsv_s,img_clara_hsv_v])
print("Imagens: Original em faixas H S V")
cv2_imshow(img_original_hsv_concat)
print("Imagens: Escura em faixas H S V")
cv2_imshow(img_escura_hsv_concat)
print("Imagens: Clara em faixas H S V")
cv2_imshow(img_clara_hsv_concat)
```

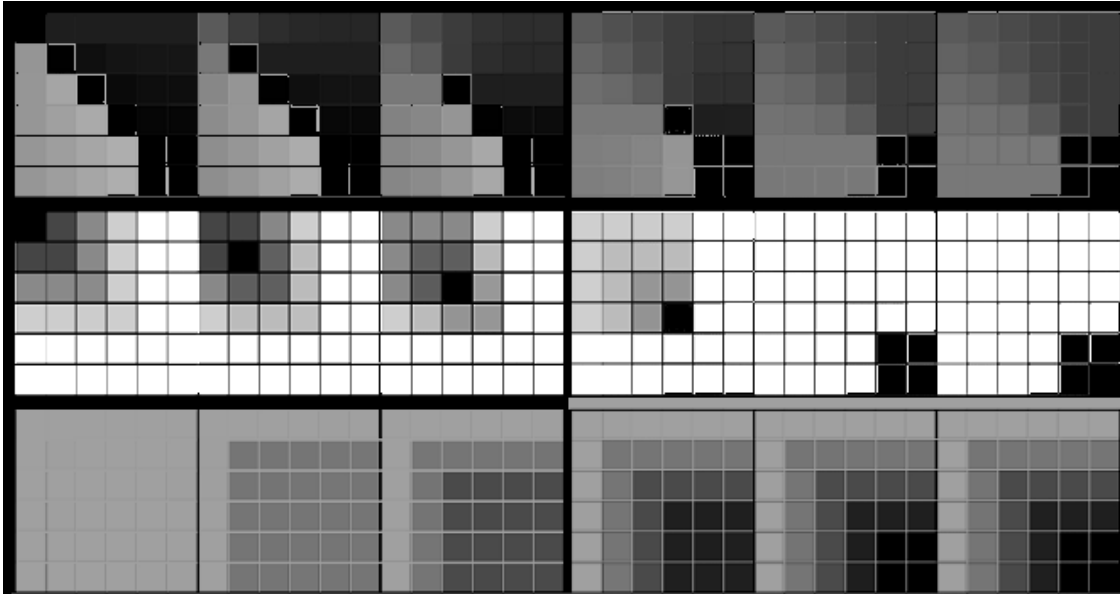
Imagens: Original + Escura + Clara



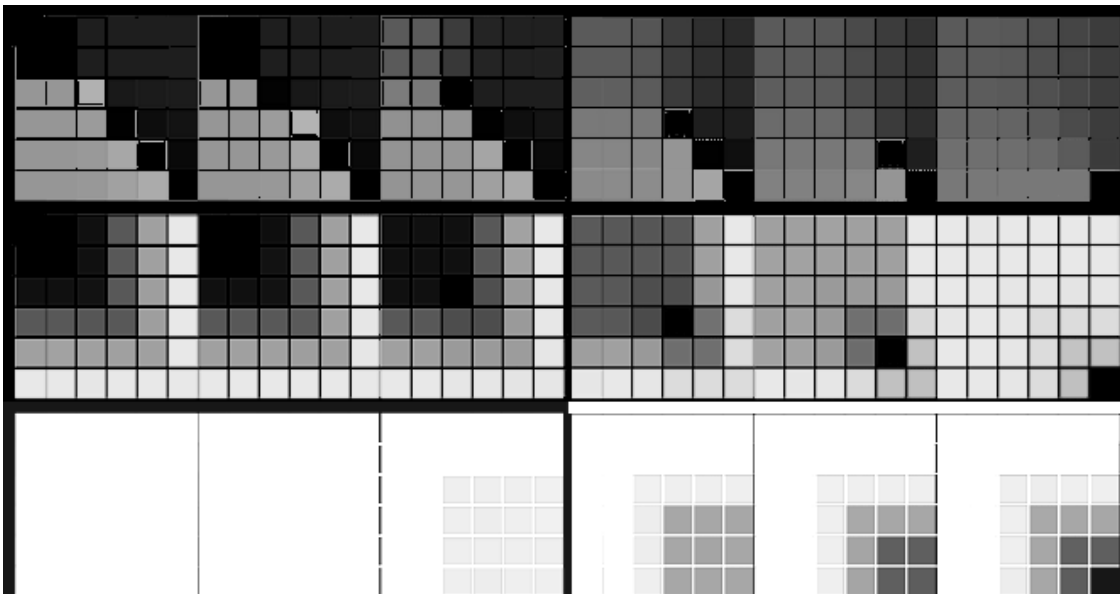
Imagens: Original em faixas H S V



Imagens: Escura em faixas H S V



Imagens: Clara em faixas H S V



Com este exercício podemos visualizar no espaço de cores HSV como a modificação no brilho e contraste da imagem modifica cada uma das bandas, dando destaque para a banda V que tem a maior mudança.

2.4 Exercício 4

Utilizamos o Script abaixo num arquivo .py para destacarmos os times do Cruzeiro e Internacional do video `cruXinter.mp4`. Realizamos o procedimento que consiste em aplicar em cada frame do vídeo uma mascara azul:

1. Menor cor azul distiguída = [241,150,70]
2. Maior cor azul distiguída = [300,200,200]

E também aplicamos uma máscara vermelha:

1. Menor cor vermelha distiguída = [0,0,70]
2. Maior cor vermelha distiguída = [60,70,255]

Após aplicar cada máscara ao vídeo original, fazemos 3 dilatações para remover os ruídos do vídeo e manter em destaque a camisa de ambos os times. Por fim, aplicamos as duas máscaras, já dilatas ao frame original e obtemos o resultado mostrado no vídeo `saída.avi`.

```
import cv2
import numpy as np

cap = cv2.VideoCapture("images/cruXinter.mp4")
# ret , frame = cap.read()
# frame_width = int(cap.get(3))
# frame_height = int(cap.get(4))
# print(frame_width, frame_height)

while(True):

    # Take each frame
    ret , frame = cap.read()
    frame_width = int(cap.get(3))
    frame_height = int(cap.get(4))
    vid_writer = cv2.VideoWriter('saida.avi',cv2.VideoWriter_fourcc('M','J','P','G'), 60, (frame_width, frame_height))

    if ret == True:
        # Convert BGR to HSV
        hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

        kernel = np.ones((5, 5), np.uint8)

        # lower_blue = np.array([177, 76, 52])
        # upper_blue = np.array([218, 114, 93])

        lower_blue = np.array([241,150,70])
        upper_blue = np.array([300,200,200])
        mask = cv2.inRange(frame, lower_blue, upper_blue)
        mask2=cv2.dilate(mask, kernel, 3)

        lower_red = np.array([0,0,70])
```

```

upper_red = np.array([60,70,255])

mask_red = cv2.inRange(frame, lower_red, upper_red)
mask2_red=cv2.dilate(mask_red, kernel, 3)

# Bitwise-AND mask and original image
res = cv2.bitwise_and(frame, frame, mask= (mask2 | mask2_red))

cv2.imshow('frame', frame)
# cv2.imshow('mask', mask)
# cv2.imshow('res', res)
cv2.imshow('erode/dilate', res)

vid_writer.write((res).astype(np.uint8))

k = cv2.waitKey(5) & 0xFF
if k == 27:
    cap.release()
    break
vid_writer.release()
cv2.destroyAllWindows()

```

Esse exercício mostrou como a dilatação, juntamente com a aplicação de mascaras podem identificar partes de um vídeo. Isso pode ser usado para diversas outras aplicações, usando o exemplo do vídeo, um dos técnicos poderia destacar a movimentação dos jogadores do seu time em relação aos jogadores do time adversário.

3 Conclusão

Com isso, concluímos que os exercícios propostos atuaram como grande fator de entendimento das maneiras de manipular os espaços de cores vistos em aula. Podemos notar também as diferenças entre as imagens utilizadas, nos diferentes espaços de cores, visto que ao mudar o espaço de cor de uma imagem, o objetivo é poder trabalhar melhor com a imagem no processamento e, não somente a imagem ter melhor visibilidade ao olho humano. Por fim, o grupo tentou adequar os resultados obtidos aos propostos pelo enunciado dos exercícios e classifica os resultados como satisfatórios.

```
[9]: !jupyter nbconvert --to pdf pratica.ipynb
```

```

[NbConvertApp] WARNING | pattern 'pratica.ipynb' matched no files
This application is used to convert notebook files (*.ipynb)
to various other formats.

```

```
WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.
```

Options

=====

The options below are convenience aliases to configurable class-options, as listed in the "Equivalent to" description-line of the aliases.

To see all configurable class-options for some <cmd>, use:

`<cmd> --help-all`

`--debug`

set log level to logging.DEBUG (maximize logging output)

Equivalent to: `[--Application.log_level=10]`

`--show-config`

Show the application's configuration (human-readable format)

Equivalent to: `[--Application.show_config=True]`

`--show-config-json`

Show the application's configuration (json format)

Equivalent to: `[--Application.show_config_json=True]`

`--generate-config`

generate default config file

Equivalent to: `[--JupyterApp.generate_config=True]`

`-y`

Answer yes to any questions instead of prompting.

Equivalent to: `[--JupyterApp.answer_yes=True]`

`--execute`

Execute the notebook prior to export.

Equivalent to: `[--ExecutePreprocessor.enabled=True]`

`--allow-errors`

Continue notebook execution even if one of the cells throws an error and include the error message in the cell output (the default behaviour is to abort conversion). This flag is only relevant if '`--execute`' was specified, too.

Equivalent to: `[--ExecutePreprocessor.allow_errors=True]`

`--stdin`

read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.*'

Equivalent to: `[--NbConvertApp.from_stdin=True]`

`--stdout`

Write notebook output to stdout instead of files.

Equivalent to: `[--NbConvertApp.writer_class=StdoutWriter]`

`--inplace`

Run nbconvert in place, overwriting the existing notebook (only relevant when converting to notebook format)

Equivalent to: `[--NbConvertApp.use_output_suffix=False]`

`--NbConvertApp.export_format=notebook --FilesWriter.build_directory=`

`--clear-output`

Clear output of current file and save in place, overwriting the existing notebook.

Equivalent to: `[--NbConvertApp.use_output_suffix=False]`

`--NbConvertApp.export_format=notebook --FilesWriter.build_directory=`

`--ClearOutputPreprocessor.enabled=True]`

```

--no-prompt
    Exclude input and output prompts from converted document.
    Equivalent to: [--TemplateExporter.exclude_input_prompt=True]
--TemplateExporter.exclude_output_prompt=True]
--no-input
    Exclude input cells and output prompts from converted document.
    This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True]
--TemplateExporter.exclude_input=True]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR',
'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
--to=<Unicode>
    The export format to be used, either one of the built-in formats
        ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook',
'pdf', 'python', 'rst', 'script', 'slides']
        or a dotted object name that represents the import path for an
        `Exporter` class
    Default: 'html'
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template file to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_file]
--writer=<DottedObjectName>
    Writer class used to write the
                                results of the conversion
    Default: 'FilesWriter'
    Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                results of the conversion
    Default: ''
    Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    overwrite base name use for output files.
                                can only be used when converting one notebook at a time.
    Default: ''
    Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults

```

to output to the directory of each notebook.

To recover previous default behaviour (outputting to the current working directory) use `.` as the flag value.

Default: `''`
 Equivalent to: `[--FilesWriter.build_directory]`
`--reveal-prefix=<Unicode>`
 The URL prefix for reveal.js (version 3.x).
 This defaults to the reveal CDN, but can be any url pointing to a copy of reveal.js.
 For speaker notes to work, this must be a relative path to a local copy of reveal.js: e.g., "reveal.js".
 If a relative path is given, it must be a subdirectory of the current directory (from which the server is run).
 See the usage documentation
 (<https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-html-slideshow>)
 for more details.

Default: `''`
 Equivalent to: `[--SlidesExporter.reveal_url_prefix]`
`--nbformat=<Enum>`
 The nbformat version to write.
 Use this to downgrade notebooks.
 Choices: any of [1, 2, 3, 4]
 Default: 4
 Equivalent to: `[--NotebookExporter.nbformat_version]`

Examples

The simplest way to use nbconvert is

```
> jupyter nbconvert mynotebook.ipynb
```

which will convert mynotebook.ipynb to the default format (probably HTML).

You can specify the export format with `--to`.`

Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'slides'].

```
> jupyter nbconvert --to latex mynotebook.ipynb
```

Both HTML and LaTeX support multiple output templates. LaTeX includes

'base', 'article' and 'report'. HTML includes 'basic' and 'full'.

You

can specify the flavor of the format used.

```
> jupyter nbconvert --to html --template basic mynotebook.ipynb
```

You can also pipe the output to stdout, rather than a file

```
> jupyter nbconvert mynotebook.ipynb --stdout
```

PDF is generated via latex

```
> jupyter nbconvert mynotebook.ipynb --to pdf
```

You can get (and serve) a Reveal.js-powered slideshow

```
> jupyter nbconvert myslides.ipynb --to slides --post serve
```

Multiple notebooks can be given at the command line in a couple of different ways:

```
> jupyter nbconvert notebook*.ipynb
```

```
> jupyter nbconvert notebook1.ipynb notebook2.ipynb
```

or you can specify the notebooks list in a config file, containing::

```
c.NbConvertApp.notebooks = ["my_notebook.ipynb"]
```

```
> jupyter nbconvert --config mycfg.py
```

To see all available configurables, use `--help-all`.