

Homework2_dhanushnd

September 27, 2021

1 ECE 4554/ ECE 5554 / Computer Vision

This file contains Problems 5 and 6 (the coding problems) for Homework 2. Your job is to implement/modify the sections within this notebook that are marked with "TO DO".

1.1 TO DO: Enter your Virginia Tech Username (PID) here: dhanushnd (906465121)

1.2 Honor Code reminder

This is not a "team project". Please review the Honor Code statement in the syllabus.

1.3 Submission guidelines for the coding problems (Google Colab)

1. Please verify that you have entered your Virginia Tech Username in all of the appropriate places.
2. After clicking Runtime->Run all, verify that all of your solutions are visible in this notebook.
3. Click File->Save near the top of the page to save the latest version of your notebook at Google Drive.
4. Verify that the last 2 cells have executed, creating a PDF version of this notebook at Google Drive. (Note: if you face difficulty with this step, please refer to <https://pypi.org/project/notebook-as-pdf/>)
5. Look at the PDF file and check that all of your solutions are displayed correctly there.
6. Download your notebook file and the PDF version to your laptop.
7. On your laptop, create a ZIP version of this notebook file. (Please don't include the separate data files.) Use file name Homework2_Code_USERNAME.zip, with your own Username.
8. For your PDF version, use file name Homework2_Notebook_USERNAME.pdf, with your own Username.
9. **Submit these 2 files and your PDF file for Problems 1-4 SEPARATELY to Canvas.** Do not zip them all together.

2 Environment Setup

```
[ ]: # Mount your Google Drive to this notebook
# The purpose is to allow your code to access to your files
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call `drive.mount("/content/drive", force_remount=True)`.

```
[57]: # Change the directory to your own working directory
      # Any files under your working directory are available to your code
      # TO DO: enter the name of your directory
      import os
      os.chdir('/content/drive/MyDrive')
      #os.chdir('/content/drive/MyDrive/Colab Notebooks')

[58]: # Import library modules
      import sys
      import cv2
      import numpy as np
      import matplotlib.pyplot as plt
      from PIL import Image, ImageOps # PIL is the Python Imaging Library
      from google.colab.patches import cv2_imshow #(We cannot use cv2.imshow on
      → Colab)
```

3 Problem 5: Image filtering (10 points)

Write Python/OpenCV code that will apply a linear filter to an image. Demonstrate correct operation of your code by applying filters that are commonly used for smoothing edge detection. (These are low-pass and high-pass filters, respectively.)

For this problem, do not use any OpenCV functions other basic operations for loading/saving/displaying image files. During the filtering operation, your code must access pixel values directly, probably with nested 'for' loops. (We know that OpenCV has built-in functions that could be used here, such as `cv2.filter2D` and `cv2.GaussianBlur`, but you are not allowed to use them. The purpose of this problem is for you to gain a good understanding of operations at the pixel level.)

```
[59]: # GETTING STARTED
      # Upload file wheel.png to your working directory.
      # Verify that you can input the image and convert it to grayscale format.
      # The resulting img_grayscale will be the input to the filtering operations
      → below.

      img_color = cv2.imread("Comp Vision/wheel.png", cv2.IMREAD_COLOR)
      cv2_imshow(img_color)

      print ('\n')
      img_grayscale = cv2.cvtColor(img_color, cv2.COLOR_BGR2GRAY)
      cv2_imshow(img_grayscale)
```





Write a Python function `linear_filter()` that accepts an image and a kernel as input parameters. Your function must create an output image by applying the kernel (also called "filter", "operator", "mask") to the input image. This new image is returned by your function.

```
[60]: def linear_filter(img_in, kernel):
    shape = kernel.shape
    #print(shape[0], shape[1])
    buf = int(shape[0]/2)

    #new_width = img_in.shape[1] + 4
    #new_height = img_in.shape[0] + 4
    #left = 2
    #top = 2
    #result = np.zeros((300+4, 400+4), dtype=np.float32)
    #result = Image.new(img_in.mode, (new_width, new_height), (128, 0, 64))

    #result.paste(pil_img, (left, top))
    #print(result)
    #print(result.shape[0], result.shape[1])
    #print(img_in.shape[0], img_in.shape[1])
    #padding = (20,20,10,10)
    #img_out = ImageOps.expand(img_in, padding)
    #return img_out
```

```

img_out = np.zeros((300, 400), dtype=np.float32)
for i in range (buf,img_in.shape[0]-shape[0]-1):
    for j in range (buf,img_in.shape[1]-shape[1]-1):
        sum = float(0)
        for a in range(shape[0]):
            for b in range(shape[1]):
                sum = sum + (img_in[i+a,j+b]*kernel[a,b])

img_out[i+buf,j+buf] =min(255,abs(sum))

```

'''Filter an input image by applying cross-correlation with a kernel.

Input:

*img_in: a grayscale image of any size larger than the kernel
kernel: a 2D array of floating-point values;
you may assume that this array is square,
with an odd number of rows and an odd number of columns;
use the *center* of this kernel as its point of reference for filtering.*

Output:

*img_out: an image with the same row/column size as img_in,
but each pixel is a floating-point value;
apply the kernel only at locations where it fits entirely within the
input image;
the remaining pixels (near the outside border of the output image)
must be set to zero;
for any negative values, take the absolute value;
clip the final output so that every pixel value lies in the range 0 to 255.*

TO DO: implement the function.

'''

```

return img_out # Each pixel must be of type np.float32

```

Test your `linear_filter()` function with the following commands.

```

[61]: # Here is an example smoothing filter,
# approximating a 2D Gaussian function with sigma = 1.
kernel = np.array([
    [1, 4, 7, 4, 1],
    [4, 16, 26, 16, 4],
    [7, 26, 41, 26, 7],
    [4, 16, 26, 16, 4],

```



```
[1, 4, 7, 4, 1]], dtype=np.float32) / 273.0

# Apply the smoothing filter
img_result = linear_filter(img_grayscale, kernel)

# Plot both images to make it easy to see that they are the same size
cv2_imshow(img_grayscale)
cv2_imshow(img_result)
```





Create a new kernel of size 9x9, approximating a 2D Gaussian function with $\sigma = 1.5$. Apply this kernel to `img_grayscale`, and plot the result.

(You must calculate these kernel coefficients yourself. You can use basic Python/NumPy math functions if you wish, but not any special OpenCV functions to create a kernel.)

```
[62]: # TO DO: write the code
def Gaussian(size, sigma):

    x = np.arange(0, size, 1, float)
    y = x[:,np.newaxis]

    x0 = y0 = size // 2

    return (np.exp(-0.5 * ((x-x0)**2 + (y-y0)**2) / 2* (sigma**2)))

matrix = Gaussian(9,1.5)
sum = matrix.sum()
#print(sum)
matrix = matrix/sum
print(matrix)

img_result = linear_filter(img_grayscale, matrix)
```

```
# Plot both images to make it easy to see that they are the same size  
cv2_imshow(img_grayscale)  
cv2_imshow(img_result)
```

```
[2.72692073e-09 1.39864352e-07 2.32895054e-06 1.25901877e-05  
 2.20964675e-05 1.25901877e-05 2.32895054e-06 1.39864352e-07  
 2.72692073e-09]  
[1.39864352e-07 7.17367271e-06 1.19452375e-04 6.45753438e-04  
 1.13333258e-03 6.45753438e-04 1.19452375e-04 7.17367271e-06  
 1.39864352e-07]  
[2.32895054e-06 1.19452375e-04 1.98906062e-03 1.07527601e-02  
 1.88716817e-02 1.07527601e-02 1.98906062e-03 1.19452375e-04  
 2.32895054e-06]  
[1.25901877e-05 6.45753438e-04 1.07527601e-02 5.81288720e-02  
 1.02019348e-01 5.81288720e-02 1.07527601e-02 6.45753438e-04  
 1.25901877e-05]  
[2.20964675e-05 1.13333258e-03 1.88716817e-02 1.02019348e-01  
 1.79049531e-01 1.02019348e-01 1.88716817e-02 1.13333258e-03  
 2.20964675e-05]  
[1.25901877e-05 6.45753438e-04 1.07527601e-02 5.81288720e-02  
 1.02019348e-01 5.81288720e-02 1.07527601e-02 6.45753438e-04  
 1.25901877e-05]  
[2.32895054e-06 1.19452375e-04 1.98906062e-03 1.07527601e-02  
 1.88716817e-02 1.07527601e-02 1.98906062e-03 1.19452375e-04  
 2.32895054e-06]  
[1.39864352e-07 7.17367271e-06 1.19452375e-04 6.45753438e-04  
 1.13333258e-03 6.45753438e-04 1.19452375e-04 7.17367271e-06  
 1.39864352e-07]  
[2.72692073e-09 1.39864352e-07 2.32895054e-06 1.25901877e-05  
 2.20964675e-05 1.25901877e-05 2.32895054e-06 1.39864352e-07  
 2.72692073e-09]]
```




Create two new 3x3 kernels that contain the two Sobel operators (horizontal and vertical). Remember to normalize the kernels (i.e., divide by 4).

Use your `linear_filter()` function to apply these two Sobel operators, creating two output images. Display the two output images, showing *magnitudes* of the computed pixel values.

Merge the two output images to create a third output image. Use any reasonable technique, such as pixelwise maximum, or pixelwise sum, or pixelwise magnitude.

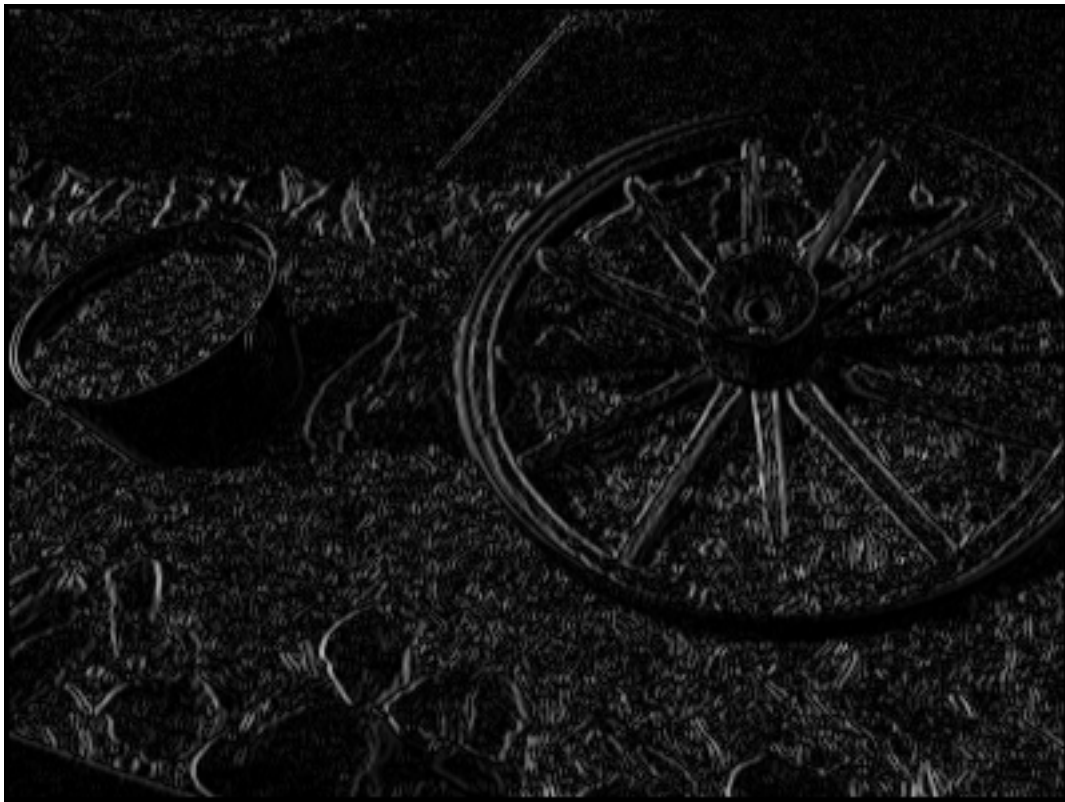
```
[63]: # TO DO: write the code
kernel_hor = np.array([[-1,0,1],[-2,0,2],[-1,0,1]], dtype=np.float32) / 4.0
img_hor = linear_filter(img_grayscale, kernel_hor)

kernel_ver = np.array([[1,2,1],[0,0,0],[-1,-2,-1]], dtype=np.float32) / 4.0
img_ver = linear_filter(img_grayscale, kernel_ver)

cv2_imshow(img_hor)
cv2_imshow(img_ver)

img_ver_and_hor = img_hor + img_ver

cv2_imshow(img_ver_and_hor)
```





4 Problem 6: Image rotation (10 points)

Write Python/OpenCV code that will input an image, and then create *rotated* versions of the input image. Image rotation is one of the geometric transformations that we discussed in a recent lecture.

For this problem, do not use any OpenCV functions other than basic operations for loading/saving/displaying image files. Your code must access pixel values directly, probably with nested 'for' loops. (Yes, OpenCV has built-in functions that could be used here, but you are not allowed to use them. The purpose of this problem is for you to gain a good understanding of operations at the pixel level.)

Hint: to avoid "holes" in your output image, write your loops to iterate over the output image, not over the input image.

```
[64]: def display_rotated_images(img_in):  
    '''Create and display rotated versions of the input image.  
  
    Input:  
        img_in: a grayscale image of any size  
  
    Output:  
        display a sequence of 6 images, each the same size as img_in;  
        initialize each output image to have pixel values of 0,  
        and then map gray values from input to output;  
        rotate about the *image center* in the counterclockwise (CCW) direction  
        by these amounts: 15, 30, 45, 60, 75, and 90 degrees.  
  
    TO DO: implement the function.  
    '''  
  
    degree = [-15, -30, -45, -60, -75, -90]  
  
    cr = img_in.shape[0]//2  
    cc = img_in.shape[1]//2  
    for d in degree:  
        d = d*np.pi/180  
        img_out = np.zeros((300, 400), dtype=np.float32)  
        '''kernel = np.array([  
            [cos(d), -sin(d), 0],  
            [sin(d), cos(d), 0],  
            [0,0,1]], dtype=np.float32)  
        '''  
        for x in range(img_out.shape[0]):  
            for y in range(img_out.shape[1]):  
                i = int(((x-cr)*np.cos(d))+((y-cc)*(-np.sin(d))))
```

```

    j = int(((x-cr)*(np.sin(d))+((y-cc)*np.cos(d))))

    i = i+cr
    j= j+cc

    if (i >= 0 and i < img_in.shape[0]) and (j >= 0 and j < img_in.
→shape[1]):
        img_out[x,y] = img_in[i,j]

    cv2_imshow(img_out)

return

```

Test your function. After showing the original image, the output should be a display of 6 images, each rotated CCW by an additional amount.

```

[65]: cv2_imshow(img_grayscale)

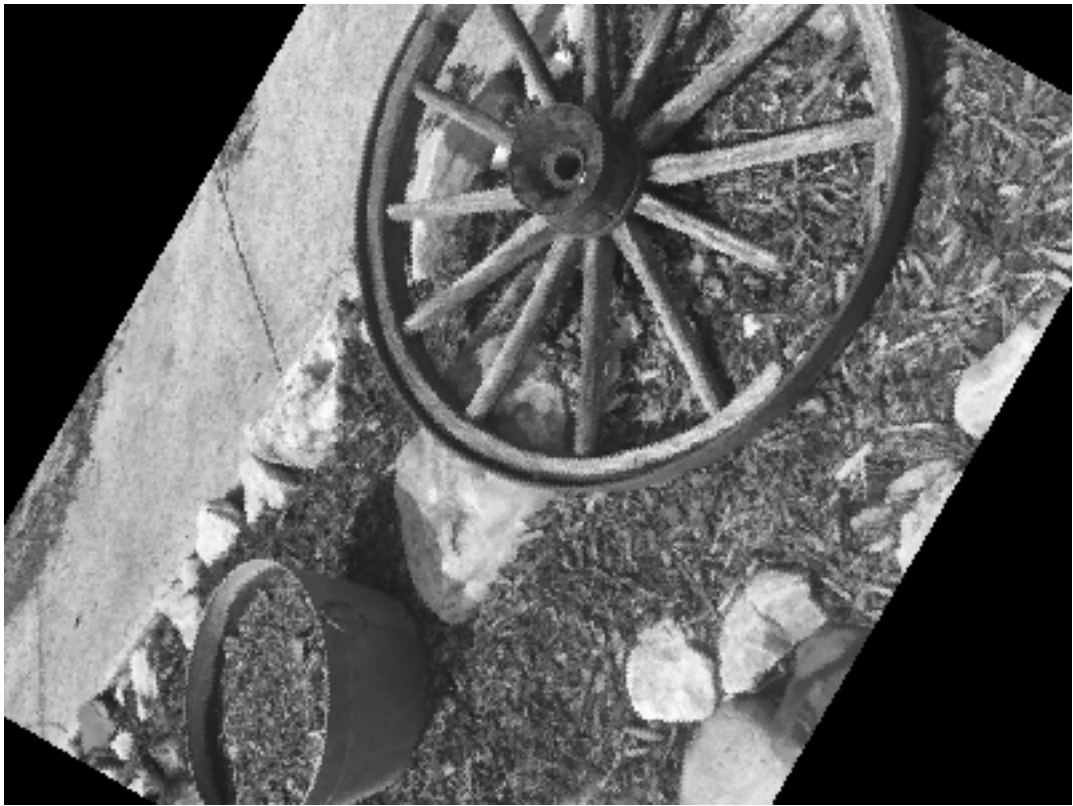
display_rotated_images(img_grayscale)

```











5 Creating a PDF version of your current notebook

```
[ ]: #The following two installation steps are needed to generate a PDF version of ↵  
      ↪the notebook  
      #(These lines are needed within Google Colab, but are not needed within a local ↵  
      ↪version of Jupyter notebook)  
      !apt-get -qq install texlive texlive-xetex texlive-latex-extra pandoc  
      !pip install --quiet py pandoc
```

```
Extracting templates from packages: 100%  
Preconfiguring packages ...  
Selecting previously unselected package fonts-droid-fallback.  
(Reading database ... 155013 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_2.2.7-1ubuntu2.8_amd64.deb ...
Unpacking libcupsimage2:amd64 (2.2.7-1ubuntu2.8) ...
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+0-0ubuntu0.18.04.14_all.deb
...
Unpacking libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.14) ...
Selecting previously unselected package libgs9:amd64.
Preparing to unpack .../13-libgs9_9.26~dfsg+0-0ubuntu0.18.04.14_amd64.deb ...
Unpacking libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.14) ...
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_2.5.1-1ubuntu1.10_amd64.deb ...
Unpacking ruby2.5 (2.5.1-1ubuntu1.10) ...
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_2.5.1-1ubuntu1.10_amd64.deb ...
Unpacking libruby2.5:amd64 (2.5.1-1ubuntu1.10) ...
Selecting previously unselected package libsyntax1:amd64.
Preparing to unpack .../28-libsyntax1_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libsyntax1: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 libtexluajit2:amd64.
Preparing to unpack
.../30-libtexluajit2_2017.20170613.44572-8ubuntu0.1_amd64.deb ...
Unpacking libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libzip-0-13:amd64.

```

```

Preparing to unpack .../31-libzip-0-13_0.13.62-3.1ubuntu0.18.04.1_amd64.deb ...
Unpacking libzip-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 tlutils.
Preparing to unpack .../34-tlutils_1.41-2_amd64.deb ...
Unpacking tlutils (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.
Preparing to unpack .../41-texlive_2017.20180305-1_all.deb ...
Unpacking texlive (2017.20180305-1) ...
Selecting previously unselected package texlive-pictures.
Preparing to unpack .../42-texlive-pictures_2017.20180305-1_all.deb ...
Unpacking texlive-pictures (2017.20180305-1) ...
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack .../43-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 .../44-texlive-plain-generic_2017.20180305-2_all.deb ...
Unpacking texlive-plain-generic (2017.20180305-2) ...
Selecting previously unselected package tipa.
Preparing to unpack .../45-tipa_2%3a1.3-20_all.deb ...
Unpacking tipa (2:1.3-20) ...
Selecting previously unselected package texlive-xetex.
Preparing to unpack .../46-texlive-xetex_2017.20180305-1_all.deb ...
Unpacking texlive-xetex (2017.20180305-1) ...

```



```
[ ]: # TO DO: Provide the full path to your Jupyter notebook file
!jupyter nbconvert --to PDF "Comp Vision/Homework2_dhanushnd.ipynb"
```