Importing Libraries

```
# Import libraries
!pip install pdf2image
!apt install poppler-utils
!apt install libtesseract-dev
!sudo apt install tesseract-ocr
!pip install pytesseract==0.3.9
!pip install tesseract

!pip install Pillow
from PIL import Image
import pytesseract
from pdf2image import convert_from_path
import cv2
import sys
import os
import csv
import numpy as np
```

```
from google.colab.patches import cv2_imshow

pytesseract.pytesseract.tesseract_cmd = r'/usr/bin/tesseract'

# Adding custom options for tesseract
# OCR Engine Mode(oem) is set to Legacy and LSTM
custom_config = r'--oem 3 --psm 6'

# Mount Google Drive
from google.colab import drive
drive.mount('/content/gdrive')
```

Functions

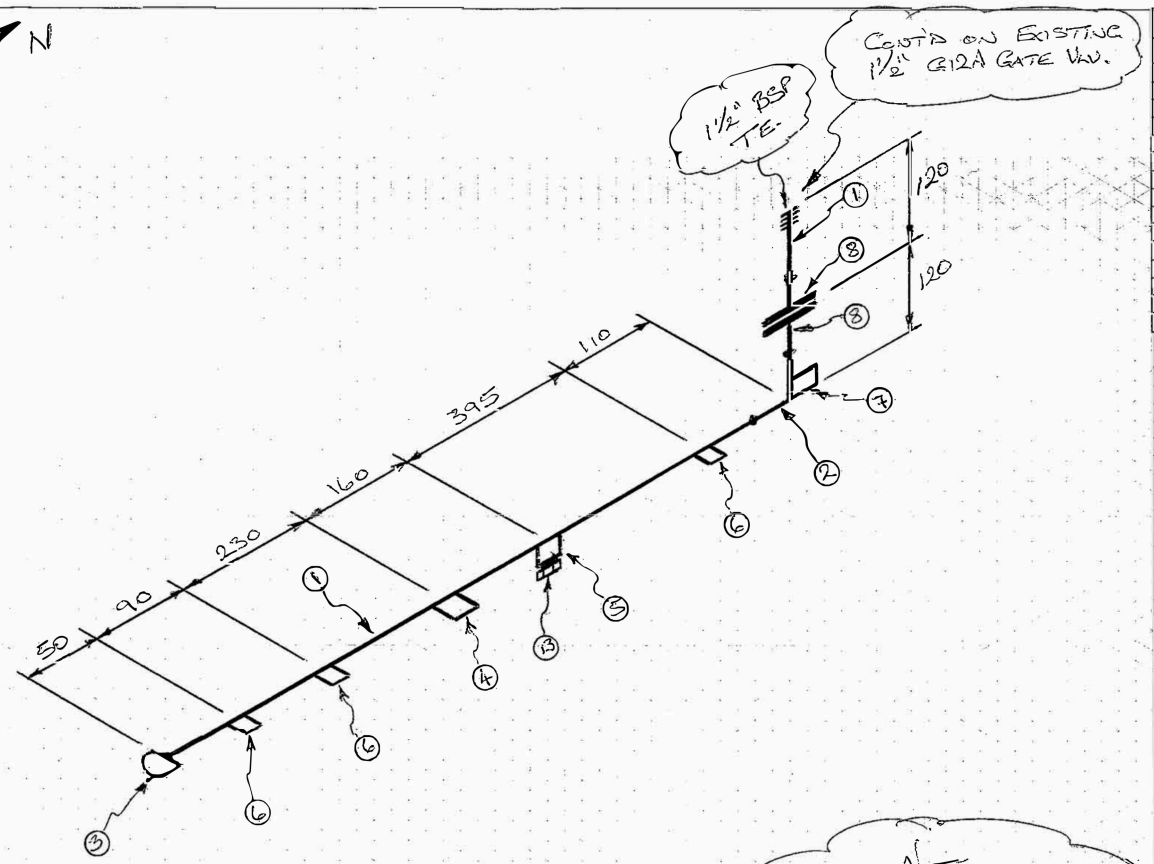
```
#  Some function for use later in code
# Tesseract image to boxes function on a 3 scale mage
def tessImage2Boxes(img):
    h, w, c = img.shape
    boxes = pytesseract.image_to_boxes(img, config=custom_config
    for b in boxes.splitlines():
        b = b.split(' ')
        img = cv2.rectangle(img, (int(b[1]), h - int(b[2])), (in
    cv2_imshow(img)

# Tesseract image to boxes function on a grayscale image
def tessImage2Boxes_gray(img):
    h, w = img.shape
    boxes = pytesseract.image_to_boxes(img, config=custom_config
    for b in boxes.splitlines():
        b = b.split(' ')
```

```
        img = cv2.rectangle(img, (int(b[1]), h - int(b[2])), (in
    cv2_imshow(img)

# Function to write jpg and text file to the
def imageToStringTextFile(imgToWrite, newFileBase, theFile):
    #print("file to write :", newFileBase + ".jpg", ", Image sha
    # get file details
    head, tail = os.path.split(theFile)
    #print("head :", head, "tail: ", tail)
    newFileName = head + "/" + newFileBase + tail[17:19]
    print("newFileName :", newFileName)

    # Write the output of the technique to a jpg
    cv2.imwrite(newFileName + ".jpg", imgToWrite)

    # Use tesseract to convert image to string - it extracts the
    text = str(((pytesseract.image_to_string(imgToWrite, lang='e
    text = text.replace('-\n\n', '')

    # # Finally, write the processed text to the file.
```

```
    with open(newFileName+".txt", 'w') as writefile:
        writefile.write(text + "\n")
```

Retrieve Files Setup

```
# File Pathname to indicate where to get of file - use google dr
afile = ('/content/gdrive/MyDrive/ComputerVision/Week 5 BOM 2')

# Read the BOM jpg here
aImg = cv2.imread(afile + '/RedactedIsometric1.jpg')
cv2_imshow(aImg)
```



NOTE
REQ'D FOR THURSDAY 5TH - JAN - 2017.

BILL OF MATERIALS

	SIZE	DESCRIPTION	QTY
1	1½"	PIPE S80 C/S SMLS A106-B GLV	1.4m
2	1½"	E.L. 90° S80 BW C/S i.R A234 WPB BLK	1
3	1½"	CAP S80 BW C/S A234 WPB BLK	1
4	1"	HCPLG 3000# TH (BSP) F/S A105 BLK	1
5	¾"	HCPLG 3000# TH (BSP) F/S A105 BLK	1
6	½"	HCPLG 3000# TH (BSP) F/S A105 BLK	1
7	¾"	CPLG 3000# TH (BSP) F/S A105 BLK	1
8	1½"	FLG 150# S80 WN F/S RF A105 BLK	2
9			
10			
11	⅞"	LG M24 HEX BOLTS & NUTS GLV	4
12	1/16"	THK GASK 150# RING UGA (ESI	1
13	¾"	PLUG 3000# TH (BSP) A105 GLV	1
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			

TITLE: TEMPERED WATER TO DILUTION MIXER 5TH FLOOR LYCRA LINE # 1

REFERENCE DRAWINGS

LINE No: NONE

P&ID No: NONE

PROJECT: —

DRAWING No:

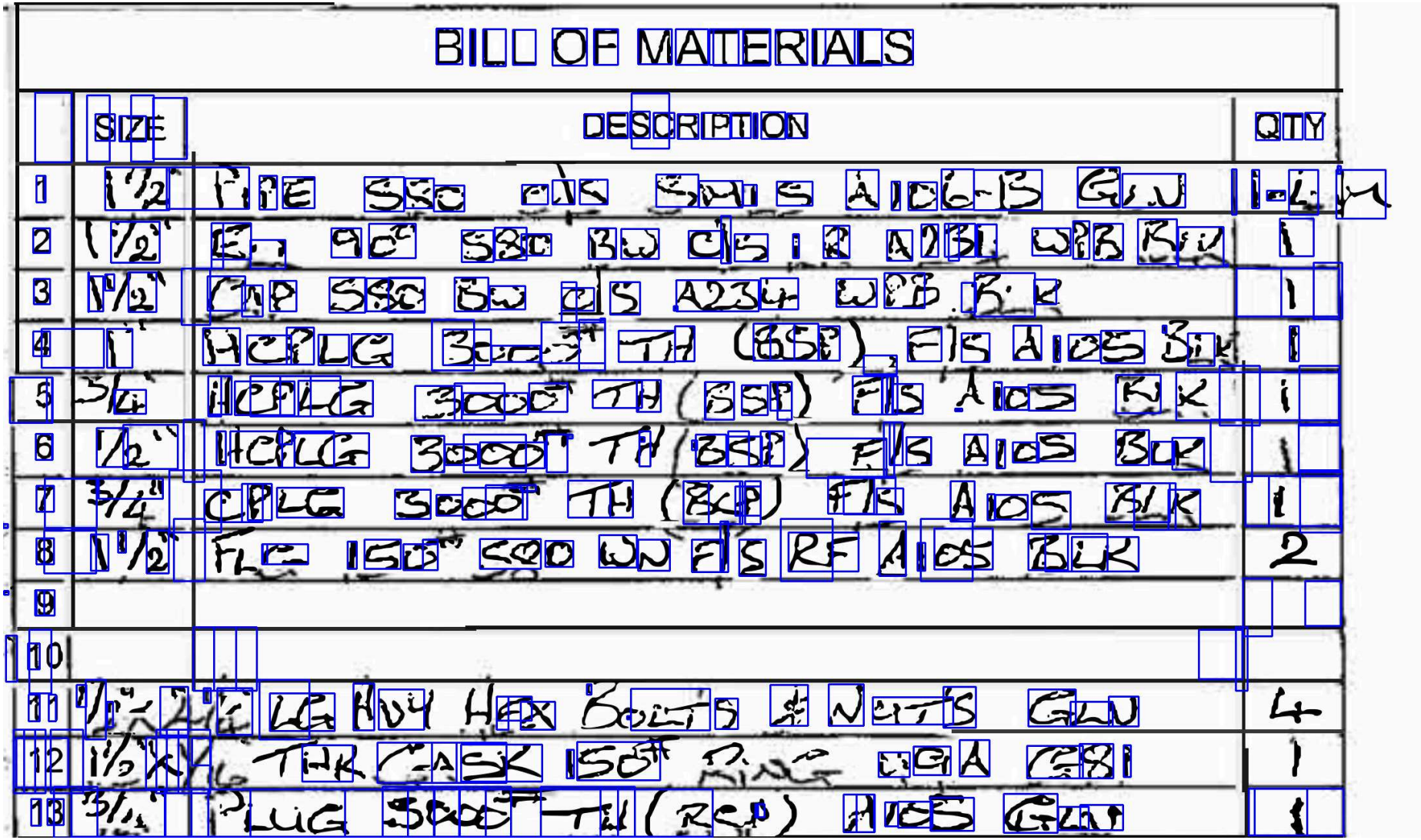
REV. 0

NOTES	LABOUR CONTENT		CHECK LIST								
	FAB	EREC	FLG. RATING:	SIZE: 1½"	SCHEDULE: S80	TYPE: C/S GLV					
			PIPE SPEC:	X-RAY: 5%	SYSTEM: TEMP WATER	HYDRO TEST PRESS: 225 PSIG					
				TRACE: NO	INSULATION: NO	TEST DURATION: 1 HOUR	0	ISSUED FOR CONSTRUCTION	MH	3-1-17	
							REV.	DESCRIPTION	APPROVED	DATE	

Extract Region of Interest

```
# Define the region of interest - ensure this is as accurate as
# Coordinates obtained from MS Paint
aImg = aImg[219:1507, 5849:8036]

# Call Function to write jpg and text file
imageToStringTextFile(aImg, "BOM_ROI_1", afile)

# Use tesseract to display image of boxes surrounding the text f
# This can be commented out as it doesn't always need to run
tessImage2Boxes(aImg)
```

newFileName : /content/gdrive/MyDrive/ComputerVision/BOM_ROI



BILL OF MATERIALS

	SIZE	DESCRIPTION	QTY
1	1/2"	PIPE SCH 80S SMLS A106-B GLV	1-4
2	1/2"	EL 90° SCH 80S BW CLS 1 A197 WPB BLK	1
3	1/2"	CAP SCH 80S BW CLS A234 WPB BLK	1
4	1"	HCPLG 3000# TH (BSP) FIS A105 BLK	1
5	3/4"	HCPLG 3000# TH (BSP) FIS A105 BLK	1
6	1/2"	HCPLG 3000# TH (BSP) FIS A105 BLK	1
7	3/4"	CPLG 3000# TH (BSP) FIS A105 BLK	1
8	1 1/2"	FLG 150# SOW WN FIS RF A105 BLK	2
9			
10			
11	1/2"-2 1/2"	LG HV4 HEX BOLTS & NUTS GLV	4
12	1 1/2"	THR CASK 150# RING CGA G5/8	1
13	3/4"	PLUG 3000# TH (RED) A105 GLV	1

Convert To GrayScale

```
img1grayscale = cv2.imread(afile + '/RedactedIsometric1.jpg')
img1grayscale = cv2.cvtColor(img1grayscale,cv2.COLOR_BGR2GRAY)
img1grayscale = img1grayscale[219:1507, 5849:8036]
```

Binary Threshold

```
# Using Simple threshold to improve the ratio rate significantly
# Highest final ratio achieved with Binary Threshold as follows.
ret, imgBin = cv2.threshold(img1grayscale,127,255,cv2.THRESH_BIN

# Call Function to write jpg and text file
imageToStringTextFile(imgBin, "BOM_BIN_1", afile)

# Use tesseract to display image of boxes surrounding the text f
```

```
# This can be commented out as it doesn't always need to run
tessImage2Boxes_gray(imgBin)
```

newFileName : /content/gdrive/MyDrive/ComputerVision/BOM_BIN

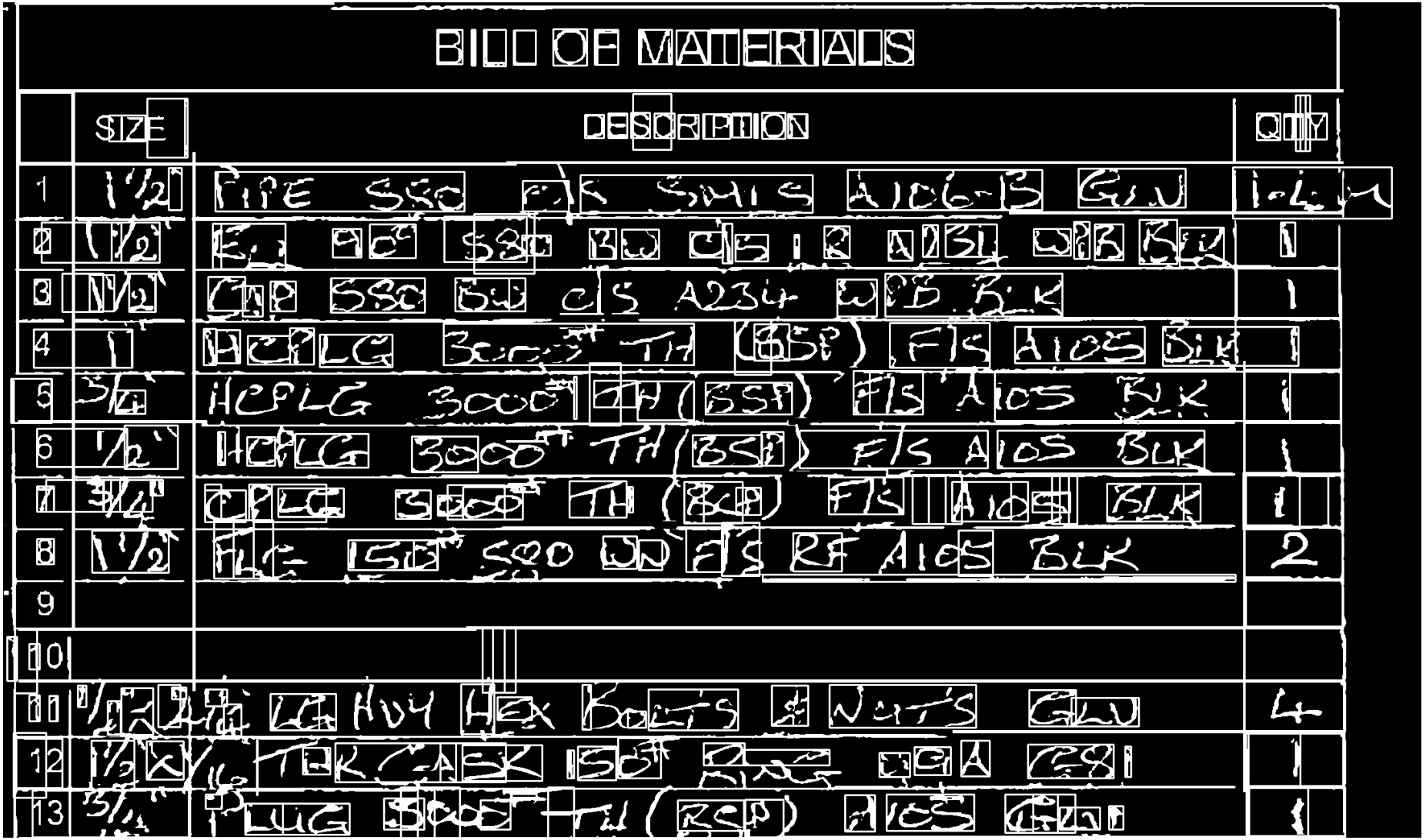
BILL OF MATERIALS

	SIZE	DESCRIPTION	QTY
1	1 1/2"	PIPE SCH 80 CS SMLS A106-B GLV	1-4
2	1/2"	EL 90° SCH BW CS LR A234 WPB BLK	1
3	1 1/2"	CAP SCH BW CS A234 WPB BLK	1
4	1"	HCPLG 3000# TH (BSP), FIS A105 BLK	1
5	3/4"	HCPLG 3000# TH (BSP) FIS A105 BLK	1
6	1/2"	HCPLG 3000# TH (BSP) FIS A105 BLK	1
7	3/4"	CPLG 3000# TH (BSP) FIS A105 BLK	1
8	1 1/2"	FLG 150# SOD WN FIS RF A105 BLK	2
9			
10			
11	1/2" X 2 1/4" LG	HVY HEX BOLTS & NUTS GLV	4
12	1/2" X 1/16"	THK GASK 150# RING DIA (SB)	1
13	3/4"	PLUG 3000# TH (RSP) A105 GLV	1

Erosion

```
# Inverse Threshold used since Erosion works better with a dark
ret, imgBinInv = cv2.threshold(img1grayscale,127,255,cv2.THRESH_

# create kernel
kernel = np.ones((2,2),np.uint8)

# Apply erosion to the image
# Spend time finding the correct balance of variables(kernel, it
imgErode = cv2.erode(imgBinInv,kernel,iterations = 1) # inverset

# Function to write jpg and text file
imageToStringTextFile(imgErode, "BOM_ERO_1", afile)

# Use tesseract to display image of boxes surrounding the text f
# This can be commented out as it doesn't always need to run
tessImage2Boxes_gray(imgErode)
```

newFileName : /content/gdrive/MyDrive/ComputerVision/BOM_ERO



BILL OF MATERIALS

	SIZE	DESCRIPTION	QTY
1	1½	PIPE SCH 80 FOR SMLS A106-B GLV	1-4
2	½	EL 90° SCH 80 BW CS A234 GR B SR	1
3	½	CAP SCH 80 BW CS A234 GR B BLK	1
4	1	HCPLG 3000 TH (BSP) FS A105 BLK	1
5	¾	HCPLG 3000 TH (BSP) FS A105 BLK	1
6	½	HCPLG 3000 TH (BSP) FS A105 BLK	1
7	¾	CPLG 3000 TH (BSP) FS A105 BLK	1
8	1½	FLG SO 300 RJ FS RF A105 BLK	2
9			
10			
11	¾ x 2¾ LG HVY HEX BOLTS & NUTS GLV		4
12	½ x 1¾ SOC CAP SCH 80 ALLOY STL		1
13	¾	PLUG SCH 80 TH (BSP) A105 GLV	1

Contouring - Horizontal and Vertical lines

```
# # making a copy of the image here as findContours might make c
imgContour = cv2.imread(afile + '/RedactedIsometric1.jpg')
imgContour = cv2.cvtColor(imgContour,cv2.COLOR_BGR2GRAY)
imgContour = imgContour[219:1507, 5849:8036]

# # Image has to be be in grayscal for the morphologyEx operatio
ret, thresh = cv2.threshold(imgContour,127,255,cv2.THRESH_BINARY

# Horizontal Kernel code here: a horizontal kernel will help to
horizontal_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3
detect_horizontal = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, hor
cnts = cv2.findContours(detect_horizontal, cv2.RETR_EXTERNAL, cv
# If third column value is NOT equal to -1 than its internal con
cnts = cnts[0] if len(cnts) == 2 else cnts[1]
for c in cnts:
  # Draw the contour
  cv2.drawContours(imgContour, [c], -1, (0,0,0), 1) # Values cha
```

```
# Vertical Kernel code here : A verticle kernel which will detec
vertical_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (2,3
detect_vertical = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, verti
cnts = cv2.findContours(detect_vertical, cv2.RETR_EXTERNAL, cv2.
# If third column value is NOT equal to -1 than its internal con
cnts = cnts[0] if len(cnts) == 2 else cnts[1]
for c in cnts:
  # Draw the contour
  cv2.drawContours(imgContour, [c], -1, (0,0,0), 1) # Values cha

# Function to write jpg and text file
imageToStringTextFile(imgContour, "BOM_HOR_VER_1", afile)

# Use tesseract to display image of boxes surrounding the text f
#tessImage2Boxes_gray(imgContour)
```

```
newFileName : /content/gdrive/MyDrive/ComputerVision/BOM_HOR
```

Erosion and Contouring

```
# apply contouring to the eroded image

# # making a copy of the image here as findContours might make c
result = imgErode.copy()
# # Image has to be be in grayscal for the morphologyEx operatio
ret, thresh = cv2.threshold(result,127,255,cv2.THRESH_BINARY)

# Horizontal Kernel
# These values will not give the best result - you are expected
# This kernel yeilded best result, trial and error
horizontal_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (1
horizontal_mask = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, horiz
cnts = cv2.findContours(horizontal_mask, cv2.RETR_EXTERNAL, cv2.
cnts = cnts[0] if len(cnts) == 2 else cnts[1]
for c in cnts:
  cv2.drawContours(result, [c], -1, (0,0,0), 1)

# Vertical Kernel
```

```
# These values will not give the best result - you are expected
# This kernel yeilded best result, trial and error
vertical_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3,1
detect_vertical = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, verti
cnts = cv2.findContours(detect_vertical, cv2.RETR_EXTERNAL, cv2.
cnts = cnts[0] if len(cnts) == 2 else cnts[1]
for c in cnts:
  cv2.drawContours(result, [c], -1, (0,0,0), 1)

# Function to write jpg and text file
imageToStringTextFile(result, "BOM_ERO_HOR_VER_1", afile)

# Use tesseract to display image of boxes surrounding the text f
#tessImage2Boxes_gray(result)
```

```
newFileName : /content/gdrive/MyDrive/ComputerVision/BOM_ERO
```

Contouring and then Erosion

```
kernel = np.ones((2,2),np.uint8)

imgContour2 = imgContour.copy()

contErode =  cv2.erode(imgContour2,kernel,iterations = 2)

# Function to write jpg and text file
imageToStringTextFile(contErode, "BOM_HOR_VER_ERO_1", afile)

# Use tesseract to display image of boxes surrounding the text f
# This can be commented out as it doesn't always need to run
#tessImage2Boxes_gray(contErode)
```

```
newFileName : /content/gdrive/MyDrive/ComputerVision/BOM_HOR
```

Blurring

```
imgBlur = cv2.imread(afile + '/RedactedIsometric1.jpg')

imgBlur = imgBlur[219:1507, 5849:8036]

imgBlur = cv2.bilateralFilter(imgBlur,d=1, sigmaColor=75, sigmaS

# Function to write jpg and text file
imageToStringTextFile(imgBlur, "BOM_BLUR_1", afile)

# Use tesseract to display image of boxes surrounding the text f
# This can be commented out as it doesn't always need to run
# tessImage2Boxes(imgBlur)
```

```
newFileName : /content/gdrive/MyDrive/ComputerVision/BOM_BLU
```

Contouring and then Blurring

```
# Applying a blur can help reduce noise, again making it easier

imgContBlur = imgContour.copy()
# use the image that has contouring applied, now apply blurring
imgContBlur = cv2.bilateralFilter(imgContBlur,d=3, sigmaColor=75

# Function to write jpg and text file
imageToStringTextFile(imgContBlur, "BOM_HOR_VER_BLUR_1", afile)

# Use tesseract to display image of boxes surrounding the text f
# This can be commented out as it doesn't always need to run
# tessImage2Boxes_gray(imgContBlur)
```

```
newFileName : /content/gdrive/MyDrive/ComputerVision/BOM_HOR
```

```
# Comparing Ground truth with the ROI
from difflib import SequenceMatcher

ratioResults = []
```

```
# Read text files
with open('/content/gdrive/MyDrive/ComputerVision/Week 5 BOM 2/
    groundTruth = f.read()

with open('/content/gdrive/MyDrive/ComputerVision/BOM_ROI_1.txt
    roi = f.read()

with open('/content/gdrive/MyDrive/ComputerVision/BOM_BIN_1.txt
```