Reading Multi Spectral Images

https://nbviewer.jupyter.org/github/thomasaarholt/hyperspy-demos/blob/master/2_SVD_and_BSS.ipynb (https://nbviewer.jupyter.org/github/thomasaarholt/hyperspy-demos/blob/master/2_SVD_and_BSS.ipynb)

Multispectral Imagery

Images obtained with a ADC Lite - Tetracam's Lightweight ADC

I made pitures about:

> Aluminum , Copper, Brass, Iron, Stainless Steel, Painted Iron

http://tetracam.com/Products-ADC_Lite.htm (http://tetracam.com/Products-ADC_Lite.htm)

MRobalinho - 11-05-2019 Version 8

Add Libraries

In [1]:

```
# Add libraries
import glob, os
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from PIL import Image, ImageFilter, ImageOps
from openpyxl import load_workbook
```

In [2]:

```
# Clear all
os.system( 'cls' )

# Verify my current folder
currDir = os.path.dirname(os.path.realpath("__file__"))
mypath = currDir
print(currDir)
```

```
C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Trabalho final\Cla
ssificacao_Sucata\Jupyter_Notebook
```

In [3]:

```
# Path to the image files
folder = "imagedata07"

# Part name of file to filter files
end_file = ".jpg"

# Upper End File
#end_file = end_file.upper()
path = currDir + "/" + folder + "/"
end_file
```

Out[3]:

```
'.jpg'
```

Read images from folder

In [4]:

```
# Read files from folder
print(path)
print('-')
print(' ---- IMAGES ON THE FOLDER :', folder, '------- *', end_file)

list_of_images = list()  # save all images on folder for further processing

for file in os.listdir(path):
    if file.endswith(end_file):
        print(os.path.join(file))
        list_of_images.append(file)   # save all images on folder for further processin
g
print('-')
```

```
C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Trabalho final\Cla
ssificacao_Sucata\Jupyter_Notebook/imagedata07/
-
 ---- IMAGES ON THE FOLDER : imagedata07 ------- * .jpg
Aluminum_1.jpg
Aluminum_2.jpg
Aluminum_3.jpg
Aluminum_4.jpg
Brass_1.jpg
Brass_2.jpg
Brass_3.jpg
CopperWire_1.jpg
CopperWire_2.jpg
CopperWire_3.jpg
Copper_1.jpg
Copper_2.jpg
Copper_3.jpg
Iron_1.jpg
Iron_2.jpg
Iron_3.jpg
PaintedIron_1.jpg
PaintedIron_2.jpg
PaintedIron_3.jpg
StainlessSteel_1.jpg
StainlessSteel_2.jpg
StainlessSteel_3.jpg
-
```

In [5]:

```
# Create Data Frame with image information
df_image = []
```

Functions to the work

In [6]:

```
# Read image with PIL
from PIL import Image, ImageFilter, ImageOps
def read_pil_image(file1):
    #print('Reading PIL image:', file1)
    try:
        im_pil = Image.open(file1)
    except:
        print("-->Unable to load image",file1)
    return im_pil
```

In [7]:

```
# Read image with OPENCV
import cv2
def read_cv2_image(file1):
    #print('Reading CV image:',file1)
    try:
        im_cv = cv2.imread(file1)
    except:
        print("-->Unable to load image",file1)
    return im_cv
```

In [8]:

```
# Look from an chanel from then image

def channel(img, n):
    """Isolate the nth channel from the image.

       n = 0: red, 1: green, 2: blue
    """
    a = np.array(img)
    a[:,:,(n!=0, n!=1, n!=2)] *= 0
#   a[:,:,n] *= 0
#   print(Image.fromarray(a), 'Get Channel n: ', n)

    print('Get Channel n: ', n)
    return Image.fromarray(a)

# def to resize
# Given parameters : image , number to divide (resize)
def imageResize(img, n):
    width, height = img.size

    print('Original size:', width, '/', height, 'Resize:',n)

    newWidth = int(width / n)
    newHeight = int(height / n)
    img.resize((newWidth, newHeight), Image.ANTIALIAS)
    print('New size:', newWidth, '/', newHeight)
    return img
```

In [9]:

```
# Obtain main color from image
# https://convertingcolors.com/rgb-color-169_171_170.html

def get_main_color(path, file):
    #img = Image.open(path+file)
    file1 = path+file
    # Read image
    img = read_pil_image(file1)
    if img == None:
        print("-->Unable to load image",file1)

    colors = img.getcolors( 1024*1024) #put a higher value if there are many colors in
 your image
    print('Get main Color file:', file)
    max_occurence, most_present = 0, 0
    try:
        for c in colors:
            if c[0] > max_occurence:
                (max_occurence, most_present) = c
        return most_present
    except TypeError:
        raise Exception("Too many colors in the image")
```

In [10]:

```
#!/usr/bin/python

# Return one 24-bit color value
def rgbToDecimal(x_rgb):
    r,g,b = rgbToRGB(x_rgb)
    rgb_dec = (r << 16) + (g << 8) + b
    #print('RGB Color:', x_rgb, '   Dec:', rgb_dec)
    return rgb_dec

# Convert 24-bit color value to RGB
def colorToRGB(c):
    r = c >> 16
    c -= r * 65536;
    g = c / 256
    c -= g * 256;
    b = c
    return [r, g, b]

def rgbToRGB(x_rgb):
    x_rgb = list(x_rgb)
    r = x_rgb[0]
    g = x_rgb[1]
    b = x_rgb[2]

    #print('rgbToRGB:',x_rgb, r,g,b)
    return r, g, b

def getRGBfromI(RGBint):
    blue =  RGBint & 255
    green = (RGBint >> 8) & 255
    red =   (RGBint >> 16) & 255
    return red, green, blue

def getIfromRGB(rgb):
    red = rgb[0]
    green = rgb[1]
    blue = rgb[2]
    #print('getIfromRGB:', red, green, blue)
    RGBint = (red<<16) + (green<<8) + blue
    return RGBint

# RGB to Hex Decimal
def rgb_to_hex(rgb):
    rgb_int = bytes(rgb).hex()
    rgb_dec = '#'+str(rgb_int)
    #print('RGB :',rgb, '  Hex Dec:', rgb_dec)
    return rgb_dec

# Test
#x_rgb = (254, 250, 255)
#rgb_hex = rgb_to_hex(x_rgb)
#rgb_dec = rgbToDecimal(x_rgb)
```

In [11]:

```
# https://github.com/conda-forge/webcolors-feedstock
# conda config --add channels conda-forge
# conda install webcolors
# It is possible to list all of the versions of webcolors available on your platform wi
th:
#       conda search webcolors --channel conda-forge

# COLOR NAME
import webcolors
def get_color_name(rgb_x):
    min_colours = {}
    for key, name in webcolors.css21_hex_to_names.items():
        r_c, g_c, b_c = webcolors.hex_to_rgb(key)
        rd = (r_c - rgb_x[0]) ** 2
        gd = (g_c - rgb_x[1]) ** 2
        bd = (b_c - rgb_x[2]) ** 2
        min_colours[(rd + gd + bd)] = name
    print('Color name from RGB:',rgb_x,'  is :',min_colours[min(min_colours.keys())])
    return min_colours[min(min_colours.keys())]
```

In [12]:

```
# Get color name from RGB
# https://stackoverflow.com/questions/2453344/find-the-colour-name-from-a-hexadecimal-colour-code

colorof = {'#F0F8FF':"aliceblue",
'#FAEBD7':"antiquewhite",
'#00FFFF':"aqua",
'#7FFFD4':"aquamarine",
'#F0FFFF':"azure",
'#F5F5DC':"beige",
'#FFE4C4':"bisque",
'#000000':"black",
'#FFEBCD':"blanchedalmond",
'#0000FF':"blue",
'#8A2BE2':"blueviolet",
'#A52A2A':"brown",
'#DEB887':"burlywood",
'#5F9EA0':"cadetblue",
'#7FFF00':"chartreuse",
'#D2691E':"chocolate",
'#FF7F50':"coral",
'#6495ED':"cornflowerblue",
'#FFF8DC':"cornsilk",
'#DC143C':"crimson",
'#00FFFF':"cyan",
'#00008B':"darkblue",
'#008B8B':"darkcyan",
'#B8860B':"darkgoldenrod",
'#A9A9A9':"darkgray",
'#006400':"darkgreen",
'#BDB76B':"darkkhaki",
'#8B008B':"darkmagenta",
'#556B2F':"darkolivegreen",
'#FF8C00':"darkorange",
'#9932CC':"darkorchid",
'#8B0000':"darkred",
'#E9967A':"darksalmon",
'#8FBC8B':"darkseagreen",
'#483D8B':"darkslateblue",
'#2F4F4F':"darkslategray",
'#00CED1':"darkturquoise",
'#9400D3':"darkviolet",
'#FF1493':"deeppink",
'#00BFFF':"deepskyblue",
'#696969':"dimgray",
'#1E90FF':"dodgerblue",
'#B22222':"firebrick",
'#FFFAF0':"floralwhite",
'#228B22':"forestgreen",
'#FF00FF':"fuchsia",
'#DCDCDC':"gainsboro",
'#F8F8FF':"ghostwhite",
'#FFD700':"gold",
'#DAA520':"goldenrod",
'#808080':"gray",
'#008000':"green",
'#ADFF2F':"greenyellow",
'#F0FFF0':"honeydew",
'#FF69B4':"hotpink",
```

```
'#CD5C5C':"indianred",
'#4B0082':"indigo",
'#FFFFF0':"ivory",
'#F0E68C':"khaki",
'#E6E6FA':"lavender",
'#FFF0F5':"lavenderblush",
'#7CFC00':"lawngreen",
'#FFFACD':"lemonchiffon",
'#ADD8E6':"lightblue",
'#F08080':"lightcoral",
'#E0FFFF':"lightcyan",
'#FAFAD2':"lightgoldenrodyellow",
'#D3D3D3':"lightgray",
'#90EE90':"lightgreen",
'#FFB6C1':"lightpink",
'#FFA07A':"lightsalmon",
'#20B2AA':"lightseagreen",
'#87CEFA':"lightskyblue",
'#778899':"lightslategray",
'#B0C4DE':"lightsteelblue",
'#FFFFE0':"lightyellow",
'#00FF00':"lime",
'#32CD32':"limegreen",
'#FAF0E6':"linen",
'#FF00FF':"magenta",
'#800000':"maroon",
'#66CDAA':"mediumaquamarine",
'#0000CD':"mediumblue",
'#BA55D3':"mediumorchid",
'#9370DB':"mediumpurple",
'#3CB371':"mediumseagreen",
'#7B68EE':"mediumslateblue",
'#00FA9A':"mediumspringgreen",
'#48D1CC':"mediumturquoise",
'#C71585':"mediumvioletred",
'#191970':"midnightblue",
'#F5FFFA':"mintcream",
'#FFE4E1':"mistyrose",
'#FFE4B5':"moccasin",
'#FFDEAD':"navajowhite",
'#000080':"navy",
'#FDF5E6':"oldlace",
'#808000':"olive",
'#6B8E23':"olivedrab",
'#FFA500':"orange",
'#FF4500':"orangered",
'#DA70D6':"orchid",
'#EEE8AA':"palegoldenrod",
'#98FB98':"palegreen",
'#AFEEEE':"paleturquoise",
'#DB7093':"palevioletred",
'#FFEFD5':"papayawhip",
'#FFDAB9':"peachpuff",
'#CD853F':"peru",
'#FFC0CB':"pink",
'#DDA0DD':"plum",
'#B0E0E6':"powderblue",
'#800080':"purple",
'#FF0000':"red",
'#BC8F8F':"rosybrown",
'#4169E1':"royalblue",
```

```
'#8B4513':"saddlebrown",
'#FA8072':"salmon",
'#F4A460':"sandybrown",
'#2E8B57':"seagreen",
'#FFF5EE':"seashell",
'#A0522D':"sienna",
'#C0C0C0':"silver",
'#87CEEB':"skyblue",
'#6A5ACD':"slateblue",
'#708090':"slategray",
'#FFFAFA':"snow",
'#00FF7F':"springgreen",
'#4682B4':"steelblue",
'#D2B48C':"tan",
'#008080':"teal",
'#D8BFD8':"thistle",
'#FF6347':"tomato",
'#40E0D0':"turquoise",
'#EE82EE':"violet",
'#F5DEB3':"wheat",
'#FFFFFF':"white",
'#F5F5F5':"whitesmoke",
'#FFFF00':"yellow",
'#9ACD32':"yellowgreen"}


def get_rgb_color_name(rgb):

    hex_from_rgb = rgb_to_hex(rgb)  # transform RGB into hexadecimal
    hx = hex_from_rgb[1:8]
    #print(hx)
    # if color is found in dict
    if colorof.get(hx):return colorof[hx]

    # else return its closest available color
    m = 16777215
    k = '000000'
    for key in colorof.keys():
        key_color = key[1:8]
        #print(key_color)
        a = int(hx[:2],16)-int(key_color[:2],16)
        b = int(hx[2:4],16)-int(key_color[2:4],16)
        c = int(hx[4:],16)-int(key_color[4:],16)

        v = a*a+b*b+c*c # simple measure for distance between colors

        # v = (r1 - r2)^2 + (g1 - g2)^2 + (b1 - b2)^2

        if v <= m:
            m = v
            k = key

    return colorof[k], hex_from_rgb

# Test
#rgb_1 = (216, 220, 223)
#cname, hexdc = get_rgb_color_name(rgb_1)
#print('Found:',    cname, '  Hex:', hexdc)     # found in dict
```

In [13]:

```
# Increase the contrast image
# im - image
# xvalue = contrast value
# https://pillow.readthedocs.io/en/4.0.x/reference/ImageEnhance.html
from PIL import ImageEnhance
# Path + file name + numeric value to enhancement

def contrast(path, xfile, xvalue):
    print('   Enhance image:', xfile, '  Value:', xvalue)
    file1 = path + xfile
    # Read Image
    im = read_pil_image(file1)
    if im == None:
        print("-->Unable to load image",file1)

    enh = ImageEnhance.Contrast(im)
    # enh.enhance(1.0).show("30% more contrast")
    x_enh = enh.enhance(xvalue)
     # Create name file masked
    f2_file = 'Enh_' + xfile
    print('   Save enhanced file :', f2_file)
    x_enh.save(f2_file)  # save enhanced file
    return x_enh, f2_file
```

In [14]:

```
# Return RGB separately
def return_rgb_from_RGB(rgb):
    p_rgb = list(rgb)
    red   = p_rgb[0]
    green = p_rgb[1]
    blue  = p_rgb[2]
    return red, green, blue
```

In [15]:

```
# Return distance from 2 colors

# http://hanzratech.in/2015/01/16/color-difference-between-2-colors-using-python.html
# https://python-colormath.readthedocs.io/en/latest/delta_e.html#delta-e-cie-2000

from colormath.color_objects import sRGBColor, LabColor
from colormath.color_conversions import convert_color
from colormath.color_diff import delta_e_cie2000

def delta_2_colors(rgb_1, rgb_2):
    #print('   Delta colors: ', rgb_1, rgb_2)
    #---- first color
    xr, xg, xb = return_rgb_from_RGB(rgb_1)
    # Red Color
    color1_rgb = sRGBColor(xr, xg, xb)

    #--- other color
    rgb_1 = rgb_2
    xr, xg, xb = return_rgb_from_RGB(rgb_1)
    # Blue Color
    color2_rgb = sRGBColor(xr, xg, xb)

    # Convert from RGB to Lab Color Space
    color1_lab = convert_color(color1_rgb, LabColor)

    # Convert from RGB to Lab Color Space
    color2_lab = convert_color(color2_rgb, LabColor)

    # Find the color difference
    delta_e = delta_e_cie2000(color1_lab, color2_lab)

    #print("      The difference between the 2 color = ", delta_e)
    return delta_e
```

In [16]:

```
# Remove Background - Put red background
#https://stackoverflow.com/questions/29313667/how-do-i-remove-the-background-from-this-kind-of-image

import cv2
import numpy as np

def red_background(path, xfile):
    print('   Red background for image:', xfile)
    #== Parameters ========================================================================
    BLUR = 21
    CANNY_THRESH_1 = 10
    CANNY_THRESH_2 = 100
    MASK_DILATE_ITER = 10
    MASK_ERODE_ITER = 10
    MASK_COLOR = (0.0,0.0,1.0) # In BGR format

    #== Processing ========================================================================
    file1 = path + xfile
    #-- Read image -----------------------------------------------------------------------
    #img = cv2.imread(file1)
    # Read image
    img = read_cv2_image(file1)
    if img.any() == None:
        print("-->Unable to load image",file1)

    # Create GRAY Image
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    #-- Edge detection -------------------------------------------------------------------
    edges = cv2.Canny(gray, CANNY_THRESH_1, CANNY_THRESH_2)
    edges = cv2.dilate(edges, None)
    edges = cv2.erode(edges, None)

    #-- Find contours in edges, sort by area ---------------------------------------------
    contour_info = []
    _, contours, _ = cv2.findContours(edges, cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)
    for c in contours:
        contour_info.append((
            c,
            cv2.isContourConvex(c),
            cv2.contourArea(c),
        ))
    contour_info = sorted(contour_info, key=lambda c: c[2], reverse=True)
    max_contour = contour_info[0]

    #-- Create empty mask, draw filled polygon on it corresponding to largest contour ----
    # Mask is black, polygon is white
    mask = np.zeros(edges.shape)
    for c in contour_info:
        cv2.fillConvexPoly(mask, c[0], (255))

    #-- Smooth mask, then blur it
```

```
    mask = cv2.dilate(mask, None, iterations=MASK_DILATE_ITER)
    mask = cv2.erode(mask, None, iterations=MASK_ERODE_ITER)
    mask = cv2.GaussianBlur(mask, (BLUR, BLUR), 0)
    mask_stack = np.dstack([mask]*3)    # Create 3-channel alpha mask

    #-- Blend masked img into MASK_COLOR background
    mask_stack  = mask_stack.astype('float32') / 255.0
    img         = img.astype('float32') / 255.0
    masked = (mask_stack * img) + ((1-mask_stack) * MASK_COLOR)
    masked = (masked * 255).astype('uint8')

    cv2.imwrite(path+"MASK_"+xfile,masked)

    # Create name file masked
    f2_file = 'Mask_'+ xfile
    file2 = path + f2_file

    # Write masked image on disk
    print('   Save masked image with red background:', f2_file)
    cv2.imwrite(file2, masked)           # Save
    # Return name file masked and image masked
    return f2_file, masked

# Test
'''
xfile = 'Brass_001.tif'
f2_file, masked = red_background(path,xfile)
%matplotlib inline
plt.imshow(masked)
plt.title('Remove image background:'+xfile,fontsize=20)
plt.show()
'''
```

Out[16]:

```
"\nxfile = 'Brass_001.tif'\nf2_file, masked = red_background(path,xfile)
\n%matplotlib inline\nplt.imshow(masked)\nplt.title('Remove image backgrou
nd:'+xfile,fontsize=20)\nplt.show()\n"
```

In [17]:

```
# https://convertingcolors.com/rgb-color-169_171_170.html

# return most_present RGB, RGB, color name, list RGB colors without RED, list RGB color
s without back

import collections

def get_main_color_without_red_and_floor(path, f2_file):
    print('    Main color from image:', f2_file)
    file1 = path + f2_file

    # Read image
    img = read_pil_image(file1)
    if img == None:
        print("-->Unable to load image",file1)

    colors = img.getcolors( 1024*1024) #put a higher value if there are many colors in
 your image
    #-----
    # Create list with colors without Background red color (near Background color)
    list_non_back = list()
    list_dec_back = list()   # List from decimal colors to list_non_back
    #
    print('...  List without excluded colors')
    # Convert list to decimal color
    for color in colors:
        # Diference between colors
       # print(color[1])
        rgb = color[1]

        excluded_rgb = False

        #Verify color name
        xt_color_name , hexdc = get_rgb_color_name(rgb)

        # Exclusion for some colors (Red Backgroud, Black foor, etc)
        if "red"   in xt_color_name:
             excluded_rgb = True
        if "black" in xt_color_name:
             excluded_rgb = True
        if "white" in xt_color_name:
            excluded_rgb = True
        if "cream" in xt_color_name:
            excluded_rgb = True

        # Force Only for non-tif files we do not delete anything
        if file.endswith('.tif'):
            excluded_rgb = False

        if  excluded_rgb == True:     # Exclude COLOR
            #print("Cor excluida", rgb, xt_color_name )
            excluded_rgb = True
        else:
            # OK COLOR - Save color in the list of correct colors (list_non_back)
            #print("Cor OK", rgb, xt_color_name )
            list_non_back.append(rgb)
            # Decimal color
            rgb_dec    = rgbToDecimal(rgb)
            list_dec_back.append(rgb_dec)
```

```
    #-----
    print('Count ocurrencies for color')
    most_present = 0

    # Most common color in the list - list_non_back
    x = collections.Counter(list_non_back)
    print('      4 Most common colors:', x.most_common(4))  # Five most common colors
    most_present = x.most_common(1)
    xrgb = list_non_back[0] # common color

    # ----- color name --
    #xt_color_name = get_color_name(xrgb)
    print('      Read color name:', xrgb)  # Color name from RGB
    xt_color_name , hexdc = get_rgb_color_name(xrgb)
    print('      Main Color file:', f2_file, ' RGB:', most_present, xrgb, ' Color nam
e:', xt_color_name,' Hex:',hexdc)

    return most_present, xrgb, xt_color_name, list_non_back, list_dec_back

# Test
#xfile = 'Copper_001.tif'
#most_present, xrgb, xt_color_name, list_non_back, \
#     list_dec_back = get_main_color_without_red_and_floor(path, xfile)
```

In [18]:

```
# https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_histograms/py_histogram_begins/py_histogram_begins.html
# Print histogram using Opencv
import cv2
import numpy as np
from matplotlib import pyplot as plt

def print_cv_hist(path, xfile):
    file1 = path + xfile
    print('Cv2 Hist from file:', file1)

    # Read image
    img_cv = read_cv2_image(file1)
    if img_cv.any() == None:
        print("-->Unable to load image",file1)

    # create a mask
    mask = np.zeros(img_cv.shape[:2], np.uint8)

    # define area to extract image from original
    #    Left:height , right:length
    mask[200:1400, 200:1800] = 255
    masked_img = cv2.bitwise_and(img_cv, img_cv ,mask = mask)

    # Calculate histogram with mask and without mask
    # Check third argument for mask
    hist_full = cv2.calcHist([img_cv],[0],None,[256],[0,256])
    hist_mask = cv2.calcHist([img_cv],[0],mask,[256],[0,256])

    plt.figure(figsize=(18,5))

    plt.subplot(141), plt.imshow(img_cv, 'gray')
    plt.title("Original")

    plt.subplot(142), plt.imshow(mask,'gray')
    plt.title('Mask')

    plt.subplot(143), plt.imshow(masked_img, 'gray')
    plt.title('Masked image')


    ax=plt.subplot(144), plt.plot(hist_full), plt.plot(hist_mask)
    ax = plt.gca()
    ax.grid(True)
    plt.title('Histogram')
    plt.xlim([0,256])

    plt.suptitle('IMAGE HISTOGRAM',fontsize=18)
    plt.xlabel('Image:'+xfile,fontsize=18)
    plt.ylabel('All chanels',fontsize=10)
    plt.savefig(path+'Hist_cv2_'+xfile)   # Save Histograme Figure
    plt.show()
    return

# Test
#xfile = 'Copper_001.tif'
#print_cv_hist(path, xfile)
```

In [19]:

```
# https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_histograms/py_histogram_begins/py_histogram_begins.html
# Print histogram using Opencv and matplotlib

import cv2
import numpy as np
from matplotlib import pyplot as plt

def print_matplot_hist(path, xfile):
    file1 = path + xfile
    print('Matplot Hist from file:', file1)

    # Read image
    img_mp = read_cv2_image(file1)
    if img_mp.any() == None:
        print("-->Unable to load image",file1)

    color = ('b','g','r')
    ax = plt.figure(figsize=(10,5))
    ax = plt.gca()
    ax.grid(True)

    for i,col in enumerate(color):
        histr = cv2.calcHist([img_mp],[i],None,[256],[0,256])
        plt.plot(histr,color = col, label='Band '+col.upper())
        plt.xlim([0,256])

    plt.title('Histogram of the image',fontsize=20)
    plt.xlabel('Image:'+xfile,fontsize=18)
    plt.ylabel('All chanels',fontsize=18)
    plt.legend(bbox_to_anchor=(.90,0.85),bbox_transform=plt.gcf().transFigure)
    plt.savefig(path+'Hist_'+xfile)   # Save Histograme Figure
    plt.show()

    return

# Test
#xfile = 'Copper_1.tif'
#print_matplot_hist(path, xfile)
```

In [20]:

```
# Max and Min value from Histogram and each position
#l = np.array(hist_full).tolist()  - Transform array in a list

import cv2
import numpy as np
from matplotlib import pyplot as plt

def histogram_max_min(path, xfile):

    file1 = path+xfile
    print('Histogram analisys:', file1)

    # Read image
    imgh = read_cv2_image(file1)
    if imgh.any() == None:
        print("-->Unable to load image",file1)

    # Calculate histogram without mask
    hist_full = cv2.calcHist([imgh],[0],None,[256],[0,256])

    # Transform array in a list
    hist_list = np.array(hist_full).tolist()

    # Valor maximo e minimo do Histograma e sua posição
    val_max  = max(hist_list)
    xval_max = int(val_max[0])

    val_avg  =  max(hist_list)
    xval_avg = int(val_avg[0]) / len(hist_list)
    xval_avg = int(xval_avg)

    val_min  = min(hist_list)
    xval_min = int(val_min[0])

    idx_max = hist_list.index(val_max)
    idx_min = hist_list.index(val_min)

    #print("Valor Max Histograma:", xval_max, '  Posição do valor Max:', idx_max)
    #print("Valor Min Histograma:", xval_min, '  Posição do valor Min:', idx_min)
    #print("Valor Avg Histograma:", xval_avg)

    return xval_max, idx_max, xval_min, idx_min

# Test
#xfile = 'Copper_001.tif'
#_,_,_,_ = histogram_max_min(path, xfile)
```

In [21]:

```
# Read image folder
import glob, os
def get_image_folder(xfile1):
    # Path to the image files
    path = currDir + "/" + folder + "/"
    # File
    file1 = path + xfile1
    print(file1)

    return file1
```

In [22]:

```
# Obtain percentage of channels R,G,B
import matplotlib.image as mpimg
def percent_rgb(path, xfile):
    print('    RGB percent from image:', xfile)
    emptyBlue = []
    emptyGreen= []
    emptyRed= []

    all_path = path + xfile
    # Read file
    img = mpimg.imread(all_path)
    imgplot = plt.imshow(img)
    # Mean of the array of each chanel
    RGBtuple = np.array(img).mean(axis=(0,1))

    averageRed = RGBtuple[0]
    averageGreen = RGBtuple[1]
    averageBlue = RGBtuple[2]

    percentageGreen = averageGreen/(averageRed+averageGreen+averageBlue) * 100
    percentageBlue = averageBlue/(averageRed+averageGreen+averageBlue) * 100
    percentageRed = averageRed/(averageRed+averageGreen+averageBlue) * 100

    emptyBlue+=[percentageBlue]
    emptyGreen+=[percentageGreen]
    emptyRed+=[percentageRed]
    print('     ------------------------------')
    print('     Percent Red',percentageRed)
    print('     Percent Green',percentageGreen)
    print('     Percent Blue',percentageBlue)
    print('     ------------------------------')
    return percentageRed, percentageGreen, percentageBlue
```

In [23]:

```
# Print all the informations from image, and create a pandas data frame with the releva
nt information

def print_file(path, xfile):
    print('--------------------------------------------------------------------------------')
    file1 = path + xfile

    # Read image
    tif_f1 = read_pil_image(file1)
    if tif_f1 == None:
        print("-->Unable to load image",file1)

    print('Inf.File:',xfile)

    # Transform Image to array
    aArray = np.array(tif_f1)
    # Array sum
    xsum = aArray.sum() / 1000000

    # Get channel 0
    x0_channel = channel(tif_f1, 0)
    aArray = np.array(x0_channel)
    xsum_0 = aArray.sum() / 1000000

    # Get channel 1
    x1_channel = channel(tif_f1, 1)
    aArray = np.array(x1_channel)
    xsum_1 = aArray.sum() / 1000000

    # Get channel 2
    x2_channel = channel(tif_f1, 2)
    aArray = np.array(x2_channel)
    xsum_2 = aArray.sum() / 1000000

    # Histogram from image
    aHist = tif_f1.histogram()
    hsum = sum(aHist) / 100000

    # Histogram channel 0
    aHist_0 = x0_channel.histogram()
    hsum_0 = sum(aHist_0) / 100000

    # Histogram channel 1
    aHist_1 = x1_channel.histogram()
    hsum_1 = sum(aHist_1) / 100000

    # Histogram chanel 0
    aHist_2 = x2_channel.histogram()
    hsum_2 = sum(aHist_2) / 100000

    # number elements on list
    nlist = len(aHist)

    # Max and Min from Histogram
    xval_max, idx_max, xval_min, idx_min = histogram_max_min(path, xfile)

    # Percentage RGB
    perc_R, perc_G, perc_B = percent_rgb(path, xfile)
```

```
    # Get color
    # Enhancement Contrast color for better definition
    # f1_file has the file name saved enhanced
    xvalue = 2.0
    print('Enhancement color:', xfile, '  Value:',xvalue)
    x_enh, f1_file = contrast(path, xfile, xvalue)

    # Remove Background - Put red background
    # f2_file has the file name saved masked

    # Only red Background for NON tif files
    #xend_file = file.endswith('*.TIF').upper()
    if file.endswith('*.TIF'):
        f2_file = f1_file
        img_masked = tif_f1
    else:
        file1 = path+f1_file
        print('Red background:', path, f1_file)
        f2_file, img_masked = red_background(path, f1_file)

    # Get Main Color -
    print('Most common color:', path, f2_file)

    # most present color, RGB from most present color:
    # color name , Hex from rgb , list colors withour red, list colors without back, de
cimal list colors without back
    most_present, xrgb, xt_color_name, list_non_back,list_dec_back = get_main_color_wit
hout_red_and_floor(path, f2_file)

    # HEX fom most present color
    hex_color  = rgb_to_hex(xrgb)

    # Decimal from most present color
    rgb_dec    = rgbToDecimal(xrgb)
    #----
    # Get Extrems of the image
    extr_a = tif_f1.getextrema()
    # Transform tuple in a list
    extr_b = [x for sets in extr_a for x in sets]
    # Sum the list
    sum_list = sum(extr_b)
    med_extr  = sum_list / len(extr_b)
    #print('List Extremes:',extr_a,'Sum:',sum_list,'Len:', len(extr_b), 'Med:',med_ext
r)


    # Obtain name file without extension
    sample_name = os.path.basename(xfile).split('_')[0]

    # Print information
    print(sample_name,' Size:',tif_f1.size, ' Format:',tif_f1.format, ' Mode:', tif_f1.
mode)
    print('          Sum array:',xsum, ' Sum Ch 0:', xsum_0, ' Sum Ch 1:', xsum_1, ' Su
m Ch 2:', xsum_2)
    print('          Histog   :',hsum ,'  N.List elem:', nlist, ' Max:', xval_max, 'Idx
Max:', idx_max, '  Min:', xval_min, 'Idx Min:', idx_min )
    print('          Color    :',xt_color_name,'   RGB   :',xrgb, '   Hex color:', hex_
color,'  Dec Color:',rgb_dec)
    print('          Extremes :',extr_a, 'Med Extremes:',med_extr)
    print('          Percentage R:', perc_R,'  Percentage G:', perc_B, '  Percentage
 B:', perc_B)
```

```
    # insert information in a Pandas Data Frame
    df_image.append((folder, xfile, sample_name, tif_f1.size, tif_f1.format, tif_f1.mod
e ,
                     xsum, xsum_0, xsum_1, xsum_2, hsum, nlist, xt_color_name, xrgb,
hex_color,
                     rgb_dec, med_extr, xval_max, idx_max, xval_min, idx_min,
                     perc_R, perc_G, perc_B))

    return most_present, xrgb, xt_color_name, list_non_back, list_dec_back
```

Starting image analysis

In [24]:

```
# Create Data Frame with image information
df_image = []

xend_file = "*" + end_file
# change work to folder path
os.chdir(path)
print('Analysing Images from:',path, xend_file)

for file in glob.glob(xend_file):
    list_dec_back = list() # List with decimal colors in the image
    print(file)
    most_present, xrgb, xt_color_name, list_non_back, list_dec_back = print_file(path,f
ile)
```

```
Analysing Images from: C:\Users\manuel.robalinho\Google Drive\UPT_Portucal
ense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ *.j
pg
Aluminum_1.jpg
------------------------------------------------------------------------
Inf.File: Aluminum_1.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Aluminu
m_1.jpg
    RGB percent from image: Aluminum_1.jpg
     ------------------------------
     Percent Red 34.011089347977304
     Percent Green 33.31607887532929
     Percent Blue 32.67283177669341
     ------------------------------
Enhancement color: Aluminum_1.jpg   Value: 2.0
   Enhance image: Aluminum_1.jpg   Value: 2.0
   Save enhanced file : Enh_Aluminum_1.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Alumin
um_1.jpg
   Red background for image: Enh_Aluminum_1.jpg
   Save masked image with red background: Mask_Enh_Aluminum_1.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Aluminum_1.jpg
    Main color from image: Mask_Enh_Aluminum_1.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((244, 255, 255), 1), ((243, 255, 255), 1),
((242, 255, 255), 1), ((241, 255, 255), 1)]
      Read color name: (244, 255, 255)
      Main Color file: Mask_Enh_Aluminum_1.jpg  RGB: [((244, 255, 255),
1)] (244, 255, 255)  Color name: azure  Hex: #f4ffff
Aluminum  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 3795.430722  Sum Ch 0: 1290.867334  Sum Ch 1: 1264.48
8693  Sum Ch 2: 1240.074695
          Histog   : 476.16768   N.List elem: 768  Max: 520979 Idx Max: 36
Min: 121 Idx Min: 1
          Color    : azure    RGB   : (244, 255, 255)    Hex color: #f4fff
f   Dec Color: 16056319
          Extremes : ((0, 255), (0, 255), (0, 255)) Med Extremes: 127.5
          Percentage R: 34.011089347977304   Percentage G: 32.672831776693
41   Percentage B: 32.67283177669341
Aluminum_2.jpg
------------------------------------------------------------------------
Inf.File: Aluminum_2.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Aluminu
m_2.jpg
    RGB percent from image: Aluminum_2.jpg
     ------------------------------
     Percent Red 33.88817939363225
     Percent Green 33.22591671742126
     Percent Blue 32.88590388894648
```

```
     ------------------------------
Enhancement color: Aluminum_2.jpg   Value: 2.0
   Enhance image: Aluminum_2.jpg   Value: 2.0
   Save enhanced file : Enh_Aluminum_2.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Alumin
um_2.jpg
   Red background for image: Enh_Aluminum_2.jpg
   Save masked image with red background: Mask_Enh_Aluminum_2.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Aluminum_2.jpg
    Main color from image: Mask_Enh_Aluminum_2.jpg
...  List without excluded colors
Count ocurrencies for color
     4 Most common colors: [((244, 255, 255), 1), ((243, 255, 255), 1),
((242, 255, 255), 1), ((241, 255, 255), 1)]
     Read color name: (244, 255, 255)
     Main Color file: Mask_Enh_Aluminum_2.jpg  RGB: [((244, 255, 255),
1)] (244, 255, 255)  Color name: azure  Hex: #f4ffff
Aluminum  Size: (5312, 2988)  Format: JPEG  Mode: RGB
         Sum array: 4078.341415  Sum Ch 0: 1382.075655  Sum Ch 1: 1355.06
6322  Sum Ch 2: 1341.199438
         Histog   : 476.16768   N.List elem: 768  Max: 521224 Idx Max: 35
Min: 48 Idx Min: 1
         Color    : azure    RGB   : (244, 255, 255)    Hex color: #f4fff
f   Dec Color: 16056319
         Extremes : ((0, 255), (0, 255), (0, 255)) Med Extremes: 127.5
         Percentage R: 33.88817939363225   Percentage G: 32.8859038889464
8   Percentage B: 32.88590388894648
Aluminum_3.jpg
------------------------------------------------------------------------
Inf.File: Aluminum_3.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Aluminu
m_3.jpg
    RGB percent from image: Aluminum_3.jpg
     ------------------------------
     Percent Red 34.00328700463863
     Percent Green 33.168799224868685
     Percent Blue 32.82791377049268
     ------------------------------
Enhancement color: Aluminum_3.jpg   Value: 2.0
   Enhance image: Aluminum_3.jpg   Value: 2.0
   Save enhanced file : Enh_Aluminum_3.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Alumin
um_3.jpg
   Red background for image: Enh_Aluminum_3.jpg
   Save masked image with red background: Mask_Enh_Aluminum_3.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Aluminum_3.jpg
    Main color from image: Mask_Enh_Aluminum_3.jpg
...  List without excluded colors
Count ocurrencies for color
     4 Most common colors: [((244, 255, 255), 1), ((243, 255, 255), 1),
((242, 255, 255), 1), ((241, 255, 255), 1)]
```

```
      Read color name: (244, 255, 255)
      Main Color file: Mask_Enh_Aluminum_3.jpg  RGB: [((244, 255, 255),
1)] (244, 255, 255)  Color name: azure  Hex: #f4ffff
Aluminum  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 3956.096638  Sum Ch 0: 1345.202894  Sum Ch 1: 1312.18
9751  Sum Ch 2: 1298.703993
          Histog   : 476.16768   N.List elem: 768  Max: 524654 Idx Max: 36
Min: 5 Idx Min: 1
          Color    : azure    RGB   : (244, 255, 255)    Hex color: #f4fff
f   Dec Color: 16056319
          Extremes : ((0, 255), (0, 255), (0, 255)) Med Extremes: 127.5
          Percentage R: 34.00328700463863   Percentage G: 32.8279137704926
8   Percentage B: 32.82791377049268
Aluminum_4.jpg
------------------------------------------------------------------------
Inf.File: Aluminum_4.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Aluminu
m_4.jpg
    RGB percent from image: Aluminum_4.jpg
     ------------------------------
     Percent Red 34.011089347977304
     Percent Green 33.31607887532929
     Percent Blue 32.67283177669341
     ------------------------------
Enhancement color: Aluminum_4.jpg   Value: 2.0
   Enhance image: Aluminum_4.jpg   Value: 2.0
   Save enhanced file : Enh_Aluminum_4.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Alumin
um_4.jpg
   Red background for image: Enh_Aluminum_4.jpg
   Save masked image with red background: Mask_Enh_Aluminum_4.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Aluminum_4.jpg
    Main color from image: Mask_Enh_Aluminum_4.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((244, 255, 255), 1), ((243, 255, 255), 1),
((242, 255, 255), 1), ((241, 255, 255), 1)]
      Read color name: (244, 255, 255)
      Main Color file: Mask_Enh_Aluminum_4.jpg  RGB: [((244, 255, 255),
1)] (244, 255, 255)  Color name: azure  Hex: #f4ffff
Aluminum  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 3795.430722  Sum Ch 0: 1290.867334  Sum Ch 1: 1264.48
8693  Sum Ch 2: 1240.074695
          Histog   : 476.16768   N.List elem: 768  Max: 520979 Idx Max: 36
Min: 121 Idx Min: 1
          Color    : azure    RGB   : (244, 255, 255)    Hex color: #f4fff
f   Dec Color: 16056319
          Extremes : ((0, 255), (0, 255), (0, 255)) Med Extremes: 127.5
          Percentage R: 34.011089347977304   Percentage G: 32.672831776693
41   Percentage B: 32.67283177669341
Brass_1.jpg
------------------------------------------------------------------------
Inf.File: Brass_1.jpg
Get Channel n:  0
```

```
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Brass_
1.jpg
    RGB percent from image: Brass_1.jpg
     ----------------------------
     Percent Red 33.710906974211774
     Percent Green 33.54178796551411
     Percent Blue 32.74730506027411
     ----------------------------
Enhancement color: Brass_1.jpg   Value: 2.0
   Enhance image: Brass_1.jpg   Value: 2.0
   Save enhanced file : Enh_Brass_1.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Brass_
1.jpg
   Red background for image: Enh_Brass_1.jpg
   Save masked image with red background: Mask_Enh_Brass_1.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Brass_1.jpg
    Main color from image: Mask_Enh_Brass_1.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((253, 255, 127), 1), ((244, 255, 255), 1),
((243, 255, 255), 1), ((242, 255, 255), 1)]
      Read color name: (253, 255, 127)
      Main Color file: Mask_Enh_Brass_1.jpg  RGB: [((253, 255, 127), 1)]
(253, 255, 127)  Color name: khaki  Hex: #fdff7f
Brass  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 809.915985  Sum Ch 0: 1720.902454  Sum Ch 1: 1712.269
126  Sum Ch 2: 1671.711701
          Histog   : 476.16768   N.List elem: 768  Max: 431630 Idx Max: 70
Min: 5 Idx Min: 0
          Color    : khaki    RGB   : (253, 255, 127)    Hex color: #fdff7
f   Dec Color: 16646015
          Extremes : ((12, 255), (4, 255), (0, 255)) Med Extremes: 130.166
66666666666
          Percentage R: 33.710906974211774   Percentage G: 32.747305060274
11   Percentage B: 32.74730506027411
Brass_2.jpg
------------------------------------------------------------------------
Inf.File: Brass_2.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Brass_
2.jpg
    RGB percent from image: Brass_2.jpg
     ----------------------------
     Percent Red 33.854032930012416
     Percent Green 33.54491998135146
     Percent Blue 32.60104708863612
     ----------------------------
Enhancement color: Brass_2.jpg   Value: 2.0
   Enhance image: Brass_2.jpg   Value: 2.0
   Save enhanced file : Enh_Brass_2.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Brass_
```

```
2.jpg
   Red background for image: Enh_Brass_2.jpg
   Save masked image with red background: Mask_Enh_Brass_2.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Brass_2.jpg
    Main color from image: Mask_Enh_Brass_2.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((253, 255, 127), 1), ((244, 255, 255), 1),
((243, 255, 255), 1), ((242, 255, 255), 1)]
      Read color name: (253, 255, 127)
      Main Color file: Mask_Enh_Brass_2.jpg  RGB: [((253, 255, 127), 1)]
(253, 255, 127)  Color name: khaki  Hex: #fdff7f
Brass  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 795.419913  Sum Ch 0: 1723.301362  Sum Ch 1: 1707.566
316  Sum Ch 2: 1659.519531
          Histog   : 476.16768   N.List elem: 768  Max: 444648 Idx Max: 93
Min: 0 Idx Min: 0
          Color    : khaki    RGB   : (253, 255, 127)    Hex color: #fdff7
f   Dec Color: 16646015
          Extremes : ((21, 255), (13, 255), (6, 255)) Med Extremes: 134.16
666666666666
          Percentage R: 33.854032930012416   Percentage G: 32.601047088636
12   Percentage B: 32.60104708863612
Brass_3.jpg
------------------------------------------------------------------------
Inf.File: Brass_3.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Brass_
3.jpg
    RGB percent from image: Brass_3.jpg
     ----------------------------
     Percent Red 33.860282475565064
     Percent Green 33.56842471834019
     Percent Blue 32.571292806094746
     ----------------------------
Enhancement color: Brass_3.jpg   Value: 2.0
   Enhance image: Brass_3.jpg   Value: 2.0
   Save enhanced file : Enh_Brass_3.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Brass_
3.jpg
   Red background for image: Enh_Brass_3.jpg
   Save masked image with red background: Mask_Enh_Brass_3.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Brass_3.jpg
    Main color from image: Mask_Enh_Brass_3.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((253, 255, 127), 1), ((244, 255, 255), 1),
((243, 255, 255), 1), ((242, 255, 255), 1)]
      Read color name: (253, 255, 127)
      Main Color file: Mask_Enh_Brass_3.jpg  RGB: [((253, 255, 127), 1)]
(253, 255, 127)  Color name: khaki  Hex: #fdff7f
Brass  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 808.952895  Sum Ch 0: 1728.201794  Sum Ch 1: 1713.305
```

```
607  Sum Ch 2: 1662.41279
          Histog   : 476.16768   N.List elem: 768  Max: 485034 Idx Max: 91
Min: 0 Idx Min: 0
          Color    : khaki    RGB   : (253, 255, 127)    Hex color: #fdff7
f   Dec Color: 16646015
          Extremes : ((25, 255), (14, 255), (5, 255)) Med Extremes: 134.83
333333333334
          Percentage R: 33.860282475565064   Percentage G: 32.571292806094
746   Percentage B: 32.571292806094746
CopperWire_1.jpg
------------------------------------------------------------------------
Inf.File: CopperWire_1.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/CopperW
ire_1.jpg
    RGB percent from image: CopperWire_1.jpg
     ----------------------------
     Percent Red 33.758377421300175
     Percent Green 33.36485657626812
     Percent Blue 32.8767660024317
     ----------------------------
Enhancement color: CopperWire_1.jpg   Value: 2.0
   Enhance image: CopperWire_1.jpg   Value: 2.0
   Save enhanced file : Enh_CopperWire_1.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Copper
Wire_1.jpg
   Red background for image: Enh_CopperWire_1.jpg
   Save masked image with red background: Mask_Enh_CopperWire_1.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_CopperWire_1.jpg
    Main color from image: Mask_Enh_CopperWire_1.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((253, 255, 127), 1), ((244, 255, 255), 1),
((243, 255, 255), 1), ((242, 255, 255), 1)]
      Read color name: (253, 255, 127)
      Main Color file: Mask_Enh_CopperWire_1.jpg  RGB: [((253, 255, 127),
1)] (253, 255, 127)  Color name: khaki  Hex: #fdff7f
CopperWire  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 1701.825834  Sum Ch 0: 2024.420058  Sum Ch 1: 2000.82
1427  Sum Ch 2: 1971.551645
          Histog   : 476.16768   N.List elem: 768  Max: 448329 Idx Max: 25
5   Min: 3 Idx Min: 1
          Color    : khaki    RGB   : (253, 255, 127)    Hex color: #fdff7
f   Dec Color: 16646015
          Extremes : ((23, 255), (4, 255), (0, 255)) Med Extremes: 132.0
          Percentage R: 33.758377421300175   Percentage G: 32.876766002431
7   Percentage B: 32.8767660024317
CopperWire_2.jpg
------------------------------------------------------------------------
Inf.File: CopperWire_2.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/CopperW
```

```
ire_2.jpg
    RGB percent from image: CopperWire_2.jpg
     ------------------------------
     Percent Red 33.88616863282227
     Percent Green 33.34134911813454
     Percent Blue 32.77248224904319
     ------------------------------
Enhancement color: CopperWire_2.jpg   Value: 2.0
   Enhance image: CopperWire_2.jpg   Value: 2.0
   Save enhanced file : Enh_CopperWire_2.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Copper
Wire_2.jpg
   Red background for image: Enh_CopperWire_2.jpg
   Save masked image with red background: Mask_Enh_CopperWire_2.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_CopperWire_2.jpg
    Main color from image: Mask_Enh_CopperWire_2.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((253, 255, 127), 1), ((244, 255, 255), 1),
((243, 255, 255), 1), ((242, 255, 255), 1)]
      Read color name: (253, 255, 127)
      Main Color file: Mask_Enh_CopperWire_2.jpg  RGB: [((253, 255, 127),
1)] (253, 255, 127)  Color name: khaki  Hex: #fdff7f
CopperWire  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 1631.193961  Sum Ch 0: 2008.148997  Sum Ch 1: 1975.86
2114  Sum Ch 2: 1942.150146
          Histog   : 476.16768   N.List elem: 768  Max: 432782 Idx Max: 25
5   Min: 0 Idx Min: 3
          Color    : khaki    RGB   : (253, 255, 127)    Hex color: #fdff7
f   Dec Color: 16646015
          Extremes : ((26, 255), (12, 255), (0, 255)) Med Extremes: 133.83
333333333334
          Percentage R: 33.88616863282227   Percentage G: 32.7724822490431
9   Percentage B: 32.77248224904319
CopperWire_3.jpg
------------------------------------------------------------------------
Inf.File: CopperWire_3.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/CopperW
ire_3.jpg
    RGB percent from image: CopperWire_3.jpg
     ------------------------------
     Percent Red 33.57139274764851
     Percent Green 33.42566170911231
     Percent Blue 33.00294554323917
     ------------------------------
Enhancement color: CopperWire_3.jpg   Value: 2.0
   Enhance image: CopperWire_3.jpg   Value: 2.0
   Save enhanced file : Enh_CopperWire_3.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Copper
Wire_3.jpg
   Red background for image: Enh_CopperWire_3.jpg
   Save masked image with red background: Mask_Enh_CopperWire_3.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
```

```
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_CopperWire_3.jpg
    Main color from image: Mask_Enh_CopperWire_3.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((253, 255, 127), 1), ((244, 255, 255), 1),
((243, 255, 255), 1), ((242, 255, 255), 1)]
      Read color name: (253, 255, 127)
      Main Color file: Mask_Enh_CopperWire_3.jpg  RGB: [((253, 255, 127),
1)] (253, 255, 127)  Color name: khaki  Hex: #fdff7f
CopperWire  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 1636.447924  Sum Ch 0: 1991.258699  Sum Ch 1: 1982.61
4786  Sum Ch 2: 1957.541735
          Histog   : 476.16768   N.List elem: 768  Max: 415869 Idx Max: 94
Min: 0 Idx Min: 1
          Color    : khaki    RGB   : (253, 255, 127)    Hex color: #fdff7
f   Dec Color: 16646015
          Extremes : ((10, 255), (11, 255), (0, 255)) Med Extremes: 131.0
          Percentage R: 33.57139274764851   Percentage G: 33.0029455432391
7   Percentage B: 33.00294554323917
Copper_1.jpg
------------------------------------------------------------------------
Inf.File: Copper_1.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Copper_
1.jpg
    RGB percent from image: Copper_1.jpg
     ------------------------------
     Percent Red 34.28333290306264
     Percent Green 33.44611662043535
     Percent Blue 32.27055047650201
     ------------------------------
Enhancement color: Copper_1.jpg   Value: 2.0
   Enhance image: Copper_1.jpg   Value: 2.0
   Save enhanced file : Enh_Copper_1.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Copper
_1.jpg
   Red background for image: Enh_Copper_1.jpg
   Save masked image with red background: Mask_Enh_Copper_1.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Copper_1.jpg
    Main color from image: Mask_Enh_Copper_1.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((253, 255, 127), 1), ((244, 255, 255), 1),
((243, 255, 255), 1), ((242, 255, 255), 1)]
      Read color name: (253, 255, 127)
      Main Color file: Mask_Enh_Copper_1.jpg  RGB: [((253, 255, 127), 1)]
(253, 255, 127)  Color name: khaki  Hex: #fdff7f
Copper  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 1037.993365  Sum Ch 0: 1828.316657  Sum Ch 1: 1783.66
8242  Sum Ch 2: 1720.975762
          Histog   : 476.16768   N.List elem: 768  Max: 341844 Idx Max: 83
Min: 0 Idx Min: 1
          Color    : khaki    RGB   : (253, 255, 127)    Hex color: #fdff7
f   Dec Color: 16646015
```

```
          Extremes : ((18, 255), (11, 255), (0, 255)) Med Extremes: 132.33
333333333334
          Percentage R: 34.28333290306264   Percentage G: 32.2705504765020
1   Percentage B: 32.27055047650201
Copper_2.jpg
----------------------------------------------------------------------
Inf.File: Copper_2.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Copper_
2.jpg
    RGB percent from image: Copper_2.jpg
     ----------------------------
     Percent Red 34.31184059071534
     Percent Green 33.35605874759947
     Percent Blue 32.33210066168519
     ----------------------------
Enhancement color: Copper_2.jpg   Value: 2.0
   Enhance image: Copper_2.jpg   Value: 2.0
   Save enhanced file : Enh_Copper_2.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Copper
_2.jpg
   Red background for image: Enh_Copper_2.jpg
   Save masked image with red background: Mask_Enh_Copper_2.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Copper_2.jpg
    Main color from image: Mask_Enh_Copper_2.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((253, 255, 191), 1), ((244, 255, 255), 1),
((243, 255, 255), 1), ((242, 255, 255), 1)]
      Read color name: (253, 255, 191)
      Main Color file: Mask_Enh_Copper_2.jpg  RGB: [((253, 255, 191), 1)]
(253, 255, 191)  Color name: lemonchiffon  Hex: #fdffbf
Copper  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 1393.597871  Sum Ch 0: 1951.851412  Sum Ch 1: 1897.48
1139  Sum Ch 2: 1839.232616
          Histog   : 476.16768   N.List elem: 768  Max: 381793 Idx Max: 87
Min: 0 Idx Min: 0
          Color    : lemonchiffon    RGB   : (253, 255, 191)    Hex color:
#fdffbf   Dec Color: 16646079
          Extremes : ((12, 255), (0, 255), (1, 255)) Med Extremes: 129.666
66666666666
          Percentage R: 34.31184059071534   Percentage G: 32.3321006616851
9   Percentage B: 32.33210066168519
Copper_3.jpg
----------------------------------------------------------------------
Inf.File: Copper_3.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Copper_
3.jpg
    RGB percent from image: Copper_3.jpg
     ----------------------------
     Percent Red 33.82658899595277
```

```
      Percent Green 33.41545561031366
      Percent Blue 32.757955393733575
      ------------------------------
Enhancement color: Copper_3.jpg   Value: 2.0
   Enhance image: Copper_3.jpg   Value: 2.0
   Save enhanced file : Enh_Copper_3.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Copper
_3.jpg
   Red background for image: Enh_Copper_3.jpg
   Save masked image with red background: Mask_Enh_Copper_3.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Copper_3.jpg
    Main color from image: Mask_Enh_Copper_3.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((253, 255, 191), 1), ((244, 255, 255), 1),
((243, 255, 255), 1), ((242, 255, 255), 1)]
      Read color name: (253, 255, 191)
      Main Color file: Mask_Enh_Copper_3.jpg  RGB: [((253, 255, 191), 1)]
(253, 255, 191)  Color name: lemonchiffon  Hex: #fdffbf
Copper  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 1339.507502  Sum Ch 0: 1905.950632  Sum Ch 1: 1882.78
5425  Sum Ch 2: 1845.738741
          Histog   : 476.16768   N.List elem: 768  Max: 347280 Idx Max: 88
Min: 0 Idx Min: 0
          Color    : lemonchiffon    RGB   : (253, 255, 191)    Hex color:
#fdffbf   Dec Color: 16646079
          Extremes : ((16, 255), (6, 255), (3, 255)) Med Extremes: 131.666
66666666666
          Percentage R: 33.82658899595277   Percentage G: 32.7579553937335
75   Percentage B: 32.757955393733575
Iron_1.jpg
------------------------------------------------------------------------
Inf.File: Iron_1.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Iron_1.
jpg
    RGB percent from image: Iron_1.jpg
     ------------------------------
     Percent Red 33.31014031631649
     Percent Green 33.466939716728774
     Percent Blue 33.22291996695473
     ------------------------------
Enhancement color: Iron_1.jpg   Value: 2.0
   Enhance image: Iron_1.jpg   Value: 2.0
   Save enhanced file : Enh_Iron_1.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Iron_
1.jpg
   Red background for image: Enh_Iron_1.jpg
   Save masked image with red background: Mask_Enh_Iron_1.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Iron_1.jpg
    Main color from image: Mask_Enh_Iron_1.jpg
...  List without excluded colors
```

```
Count ocurrencies for color
      4 Most common colors: [((253, 255, 127), 1), ((244, 255, 255), 1),
((243, 255, 255), 1), ((242, 255, 255), 1)]
      Read color name: (253, 255, 127)
      Main Color file: Mask_Enh_Iron_1.jpg  RGB: [((253, 255, 127), 1)] (2
53, 255, 127)  Color name: khaki  Hex: #fdff7f
Iron  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 290.375493  Sum Ch 0: 1527.384117  Sum Ch 1: 1534.573
907  Sum Ch 2: 1523.384765
          Histog   : 476.16768   N.List elem: 768  Max: 464308 Idx Max: 57
Min: 15 Idx Min: 1
          Color    : khaki    RGB   : (253, 255, 127)    Hex color: #fdff7
f   Dec Color: 16646015
          Extremes : ((0, 255), (0, 255), (0, 255)) Med Extremes: 127.5
          Percentage R: 33.31014031631649   Percentage G: 33.2229199669547
3   Percentage B: 33.22291996695473
Iron_2.jpg
----------------------------------------------------------------------
Inf.File: Iron_2.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Iron_2.
jpg
    RGB percent from image: Iron_2.jpg
     ------------------------------
     Percent Red 33.31725431451083
     Percent Green 33.57483981608991
     Percent Blue 33.10790586939924
     ------------------------------
Enhancement color: Iron_2.jpg   Value: 2.0
   Enhance image: Iron_2.jpg   Value: 2.0
   Save enhanced file : Enh_Iron_2.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Iron_
2.jpg
   Red background for image: Enh_Iron_2.jpg
   Save masked image with red background: Mask_Enh_Iron_2.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Iron_2.jpg
    Main color from image: Mask_Enh_Iron_2.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((253, 255, 191), 1), ((244, 255, 255), 1),
((243, 255, 255), 1), ((242, 255, 255), 1)]
      Read color name: (253, 255, 191)
      Main Color file: Mask_Enh_Iron_2.jpg  RGB: [((253, 255, 191), 1)] (2
53, 255, 191)  Color name: lemonchiffon  Hex: #fdffbf
Iron  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 563.473852  Sum Ch 0: 1618.699193  Sum Ch 1: 1631.213
833  Sum Ch 2: 1608.528122
          Histog   : 476.16768   N.List elem: 768  Max: 447629 Idx Max: 62
Min: 5 Idx Min: 2
          Color    : lemonchiffon    RGB   : (253, 255, 191)    Hex color:
#fdffbf   Dec Color: 16646079
          Extremes : ((0, 255), (1, 255), (0, 255)) Med Extremes: 127.6666
6666666667
          Percentage R: 33.31725431451083   Percentage G: 33.1079058693992
4   Percentage B: 33.10790586939924
```

```
Iron_3.jpg
------------------------------------------------------------------------
Inf.File: Iron_3.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Iron_3.
jpg
    RGB percent from image: Iron_3.jpg
     ------------------------------
     Percent Red 33.213108214862395
     Percent Green 33.79137679279476
     Percent Blue 32.99551499234284
     ------------------------------
Enhancement color: Iron_3.jpg   Value: 2.0
   Enhance image: Iron_3.jpg   Value: 2.0
   Save enhanced file : Enh_Iron_3.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Iron_
3.jpg
   Red background for image: Enh_Iron_3.jpg
   Save masked image with red background: Mask_Enh_Iron_3.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_Iron_3.jpg
    Main color from image: Mask_Enh_Iron_3.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((253, 255, 127), 1), ((244, 255, 255), 1),
((243, 255, 255), 1), ((242, 255, 255), 1)]
      Read color name: (253, 255, 127)
      Main Color file: Mask_Enh_Iron_3.jpg  RGB: [((253, 255, 127), 1)] (2
53, 255, 127)  Color name: khaki  Hex: #fdff7f
Iron  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 328.077431  Sum Ch 0: 1535.456848  Sum Ch 1: 1562.190
463  Sum Ch 2: 1525.397416
          Histog   : 476.16768   N.List elem: 768  Max: 451514 Idx Max: 56
Min: 5 Idx Min: 1
          Color    : khaki    RGB   : (253, 255, 127)    Hex color: #fdff7
f   Dec Color: 16646015
          Extremes : ((2, 255), (0, 255), (0, 255)) Med Extremes: 127.8333
3333333333
          Percentage R: 33.213108214862395   Percentage G: 32.995514992342
84   Percentage B: 32.99551499234284
PaintedIron_1.jpg
------------------------------------------------------------------------
Inf.File: PaintedIron_1.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Painted
Iron_1.jpg
    RGB percent from image: PaintedIron_1.jpg
     ------------------------------
     Percent Red 33.8282320864359
     Percent Green 33.248045362405875
     Percent Blue 32.92372255115822
     ------------------------------
Enhancement color: PaintedIron_1.jpg   Value: 2.0
```

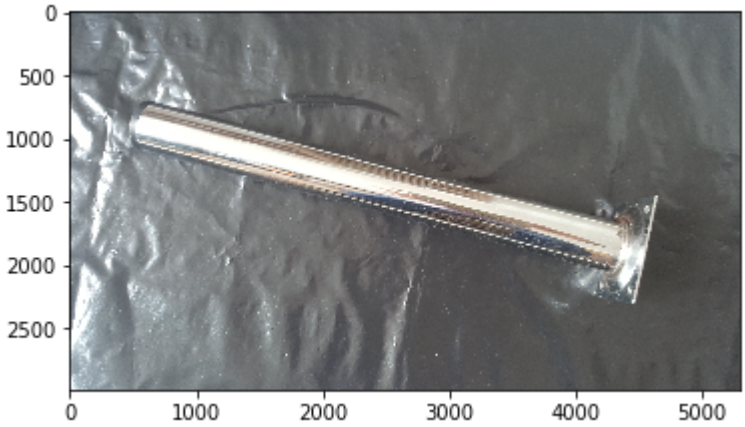
```
   Enhance image: PaintedIron_1.jpg   Value: 2.0
   Save enhanced file : Enh_PaintedIron_1.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Painte
dIron_1.jpg
   Red background for image: Enh_PaintedIron_1.jpg
   Save masked image with red background: Mask_Enh_PaintedIron_1.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_PaintedIron_1.jpg
    Main color from image: Mask_Enh_PaintedIron_1.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((244, 255, 255), 1), ((243, 255, 255), 1),
((242, 255, 255), 1), ((241, 255, 255), 1)]
      Read color name: (244, 255, 255)
      Main Color file: Mask_Enh_PaintedIron_1.jpg  RGB: [((244, 255, 255),
1)] (244, 255, 255)  Color name: azure  Hex: #f4ffff
PaintedIron  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 71.007131  Sum Ch 0: 1476.931962  Sum Ch 1: 1451.6011
58  Sum Ch 2: 1437.441307
          Histog   : 476.16768   N.List elem: 768  Max: 587224 Idx Max: 25
5   Min: 385 Idx Min: 1
          Color    : azure    RGB   : (244, 255, 255)    Hex color: #f4fff
f   Dec Color: 16056319
          Extremes : ((0, 255), (0, 255), (0, 255)) Med Extremes: 127.5
          Percentage R: 33.8282320864359   Percentage G: 32.92372255115822
Percentage B: 32.92372255115822
PaintedIron_2.jpg
------------------------------------------------------------------------
Inf.File: PaintedIron_2.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Painted
Iron_2.jpg
    RGB percent from image: PaintedIron_2.jpg
     ------------------------------
     Percent Red 33.615164055035294
     Percent Green 33.34086189211816
     Percent Blue 33.04397405284654
     ------------------------------
Enhancement color: PaintedIron_2.jpg   Value: 2.0
   Enhance image: PaintedIron_2.jpg   Value: 2.0
   Save enhanced file : Enh_PaintedIron_2.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Painte
dIron_2.jpg
   Red background for image: Enh_PaintedIron_2.jpg
   Save masked image with red background: Mask_Enh_PaintedIron_2.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_PaintedIron_2.jpg
    Main color from image: Mask_Enh_PaintedIron_2.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((244, 255, 255), 1), ((243, 255, 255), 1),
((242, 255, 255), 1), ((241, 255, 255), 1)]
      Read color name: (244, 255, 255)
      Main Color file: Mask_Enh_PaintedIron_2.jpg  RGB: [((244, 255, 255),
```

```
1)] (244, 255, 255)  Color name: azure  Hex: #f4ffff
PaintedIron  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 4287.385442  Sum Ch 0: 1441.21165  Sum Ch 1: 1429.451
259  Sum Ch 2: 1416.722533
          Histog   : 476.16768   N.List elem: 768  Max: 407549 Idx Max: 53
Min: 11 Idx Min: 1
          Color    : azure    RGB   : (244, 255, 255)    Hex color: #f4fff
f   Dec Color: 16056319
          Extremes : ((0, 255), (0, 255), (0, 255)) Med Extremes: 127.5
          Percentage R: 33.615164055035294   Percentage G: 33.043974052846
54   Percentage B: 33.04397405284654
PaintedIron_3.jpg
----------------------------------------------------------------------
Inf.File: PaintedIron_3.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Painted
Iron_3.jpg
    RGB percent from image: PaintedIron_3.jpg
     ------------------------------
     Percent Red 33.38725786908276
     Percent Green 33.4341077688277
     Percent Blue 33.178634362089554
     ------------------------------
Enhancement color: PaintedIron_3.jpg   Value: 2.0
   Enhance image: PaintedIron_3.jpg   Value: 2.0
   Save enhanced file : Enh_PaintedIron_3.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Painte
dIron_3.jpg
   Red background for image: Enh_PaintedIron_3.jpg
   Save masked image with red background: Mask_Enh_PaintedIron_3.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_PaintedIron_3.jpg
    Main color from image: Mask_Enh_PaintedIron_3.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((244, 255, 255), 1), ((243, 255, 255), 1),
((242, 255, 255), 1), ((241, 255, 255), 1)]
      Read color name: (244, 255, 255)
      Main Color file: Mask_Enh_PaintedIron_3.jpg  RGB: [((244, 255, 255),
1)] (244, 255, 255)  Color name: azure  Hex: #f4ffff
PaintedIron  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 335.612517  Sum Ch 0: 1546.023623  Sum Ch 1: 1548.193
045  Sum Ch 2: 1536.363145
          Histog   : 476.16768   N.List elem: 768  Max: 430827 Idx Max: 25
5   Min: 125 Idx Min: 1
          Color    : azure    RGB   : (244, 255, 255)    Hex color: #f4fff
f   Dec Color: 16056319
          Extremes : ((0, 255), (0, 255), (0, 255)) Med Extremes: 127.5
          Percentage R: 33.38725786908276   Percentage G: 33.1786343620895
54   Percentage B: 33.178634362089554
StainlessSteel_1.jpg
----------------------------------------------------------------------
Inf.File: StainlessSteel_1.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
```

```
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Stainle
ssSteel_1.jpg
    RGB percent from image: StainlessSteel_1.jpg
     ------------------------------
     Percent Red 33.11307992979201
     Percent Green 33.365975452247866
     Percent Blue 33.520944617960126
     ------------------------------
Enhancement color: StainlessSteel_1.jpg   Value: 2.0
   Enhance image: StainlessSteel_1.jpg   Value: 2.0
   Save enhanced file : Enh_StainlessSteel_1.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Stainl
essSteel_1.jpg
   Red background for image: Enh_StainlessSteel_1.jpg
   Save masked image with red background: Mask_Enh_StainlessSteel_1.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_StainlessSteel_1.jpg
    Main color from image: Mask_Enh_StainlessSteel_1.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((244, 255, 255), 1), ((243, 255, 255), 1),
((242, 255, 255), 1), ((241, 255, 255), 1)]
      Read color name: (244, 255, 255)
      Main Color file: Mask_Enh_StainlessSteel_1.jpg  RGB: [((244, 255, 25
5), 1)] (244, 255, 255)  Color name: azure  Hex: #f4ffff
StainlessSteel  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 1256.49319  Sum Ch 0: 1838.259548  Sum Ch 1: 1852.298
943  Sum Ch 2: 1860.901995
          Histog   : 476.16768   N.List elem: 768  Max: 323785 Idx Max: 84
Min: 0 Idx Min: 0
          Color    : azure    RGB   : (244, 255, 255)    Hex color: #f4fff
f   Dec Color: 16056319
          Extremes : ((21, 255), (16, 255), (14, 255)) Med Extremes: 136.0
          Percentage R: 33.11307992979201   Percentage G: 33.5209446179601
26   Percentage B: 33.520944617960126
StainlessSteel_2.jpg
----------------------------------------------------------------------
Inf.File: StainlessSteel_2.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Stainle
ssSteel_2.jpg
    RGB percent from image: StainlessSteel_2.jpg
     ------------------------------
     Percent Red 33.094499985169875
     Percent Green 33.40192656927938
     Percent Blue 33.50357344555074
     ------------------------------
Enhancement color: StainlessSteel_2.jpg   Value: 2.0
   Enhance image: StainlessSteel_2.jpg   Value: 2.0
   Save enhanced file : Enh_StainlessSteel_2.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Stainl
essSteel_2.jpg
   Red background for image: Enh_StainlessSteel_2.jpg
   Save masked image with red background: Mask_Enh_StainlessSteel_2.jpg
```

```
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_StainlessSteel_2.jpg
    Main color from image: Mask_Enh_StainlessSteel_2.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((244, 255, 255), 1), ((243, 255, 255), 1),
((242, 255, 255), 1), ((241, 255, 255), 1)]
      Read color name: (244, 255, 255)
      Main Color file: Mask_Enh_StainlessSteel_2.jpg  RGB: [((244, 255, 25
5), 1)] (244, 255, 255)  Color name: azure  Hex: #f4ffff
StainlessSteel  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 1159.233761  Sum Ch 0: 1805.040568  Sum Ch 1: 1821.80
8232  Sum Ch 2: 1827.352257
          Histog   : 476.16768   N.List elem: 768  Max: 390515 Idx Max: 85
Min: 0 Idx Min: 0
          Color    : azure    RGB   : (244, 255, 255)    Hex color: #f4fff
f   Dec Color: 16056319
          Extremes : ((20, 255), (13, 255), (11, 255)) Med Extremes: 134.8
3333333333334
          Percentage R: 33.094499985169875   Percentage G: 33.503573445550
74   Percentage B: 33.50357344555074
StainlessSteel_3.jpg
------------------------------------------------------------------------
Inf.File: StainlessSteel_3.jpg
Get Channel n:  0
Get Channel n:  1
Get Channel n:  2
Histogram analisys: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Stainle
ssSteel_3.jpg
    RGB percent from image: StainlessSteel_3.jpg
     ------------------------------
     Percent Red 33.05809742388418
     Percent Green 33.47459374563412
     Percent Blue 33.4673088304817
     ------------------------------
Enhancement color: StainlessSteel_3.jpg   Value: 2.0
   Enhance image: StainlessSteel_3.jpg   Value: 2.0
   Save enhanced file : Enh_StainlessSteel_3.jpg
Red background: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense\Tr
abalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Enh_Stainl
essSteel_3.jpg
   Red background for image: Enh_StainlessSteel_3.jpg
   Save masked image with red background: Mask_Enh_StainlessSteel_3.jpg
Most common color: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalense
\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/ Mask_En
h_StainlessSteel_3.jpg
    Main color from image: Mask_Enh_StainlessSteel_3.jpg
...  List without excluded colors
Count ocurrencies for color
      4 Most common colors: [((244, 255, 255), 1), ((243, 255, 255), 1),
((242, 255, 255), 1), ((241, 255, 255), 1)]
      Read color name: (244, 255, 255)
      Main Color file: Mask_Enh_StainlessSteel_3.jpg  RGB: [((244, 255, 25
5), 1)] (244, 255, 255)  Color name: azure  Hex: #f4ffff
StainlessSteel  Size: (5312, 2988)  Format: JPEG  Mode: RGB
          Sum array: 1193.991623  Sum Ch 0: 1814.545387  Sum Ch 1: 1837.40
6699  Sum Ch 2: 1837.006833
          Histog   : 476.16768   N.List elem: 768  Max: 347369 Idx Max: 82
Min: 0 Idx Min: 1
```

```
          Color    : azure    RGB    : (244, 255, 255)    Hex color: #f4fff
f   Dec Color: 16056319
          Extremes : ((0, 255), (0, 255), (0, 255)) Med Extremes: 127.5
          Percentage R: 33.05809742388418   Percentage G: 33.4673088304817
Percentage B: 33.4673088304817
```



In [25]:

```
#list_dec_back ordered
order_list_dec = sorted(list_dec_back, key=int)
#order_list_dec
#list_non_back
```

In [26]:

```
'''
TESTS
# Read all list to see the color - obtain RGB from int
for x in order_list_dec:
    #print(x)
    # Get RGB from INT
    xrgb = getRGBfromI(x)
    #print('Int:, x,'  RGB: ',xrgb)
    xt_color_name , hexdc = get_rgb_color_name(xrgb)
    print('Int:', x,'  RGB: ', xrgb, xt_color_name)
'''
```

Out[26]:

```
"\nTESTS\n# Read all list to see the color - obtain RGB from int\nfor x in
order_list_dec:\n    #print(x)\n    # Get RGB from INT\n    xrgb = getRGBf
romI(x)\n    #print('Int:, x,'  RGB: ',xrgb)\n    xt_color_name , hexdc =
get_rgb_color_name(xrgb)\n    print('Int:', x,'  RGB: ', xrgb, xt_color_na
me)\n"
```

In [27]:

```
df = pd.DataFrame(df_image,columns=['Folder','File','Material','Size','Format','Mode',
                                    'All_Bands', 'Sum_Ch0','Sum_Ch1','Sum_Ch2',
                                    'Histogram','Number_elements',
                                    'Color','Color_RGB', 'Color_hex','Color_dec','Med_E
xtrems',
                                    'Max_Histog', 'Idx_Max_Histog','Min_Histog', 'Idx_M
in_Histog',
                                    'perc_R', 'perc_G', 'perc_B'])
df.head(50)
```

Out[27]:

	Folder	File	Material	Size	Format	Mode	All_Bands	Su
0	imagedata07	Aluminum_1.jpg	Aluminum	(5312, 2988)	JPEG	RGB	3795.430722	1290.
1	imagedata07	Aluminum_2.jpg	Aluminum	(5312, 2988)	JPEG	RGB	4078.341415	1382.
2	imagedata07	Aluminum_3.jpg	Aluminum	(5312, 2988)	JPEG	RGB	3956.096638	1345.
3	imagedata07	Aluminum_4.jpg	Aluminum	(5312, 2988)	JPEG	RGB	3795.430722	1290.
4	imagedata07	Brass_1.jpg	Brass	(5312, 2988)	JPEG	RGB	809.915985	1720.
5	imagedata07	Brass_2.jpg	Brass	(5312, 2988)	JPEG	RGB	795.419913	1723.
6	imagedata07	Brass_3.jpg	Brass	(5312, 2988)	JPEG	RGB	808.952895	1728.
7	imagedata07	CopperWire_1.jpg	CopperWire	(5312, 2988)	JPEG	RGB	1701.825834	2024.
8	imagedata07	CopperWire_2.jpg	CopperWire	(5312, 2988)	JPEG	RGB	1631.193961	2008.
9	imagedata07	CopperWire_3.jpg	CopperWire	(5312, 2988)	JPEG	RGB	1636.447924	1991.
10	imagedata07	Copper_1.jpg	Copper	(5312, 2988)	JPEG	RGB	1037.993365	1828.
11	imagedata07	Copper_2.jpg	Copper	(5312, 2988)	JPEG	RGB	1393.597871	1951.
12	imagedata07	Copper_3.jpg	Copper	(5312, 2988)	JPEG	RGB	1339.507502	1905.
13	imagedata07	Iron_1.jpg	Iron	(5312, 2988)	JPEG	RGB	290.375493	1527.
14	imagedata07	Iron_2.jpg	Iron	(5312, 2988)	JPEG	RGB	563.473852	1618.
15	imagedata07	Iron_3.jpg	Iron	(5312, 2988)	JPEG	RGB	328.077431	1535.
16	imagedata07	PaintedIron_1.jpg	PaintedIron	(5312, 2988)	JPEG	RGB	71.007131	1476.
17	imagedata07	PaintedIron_2.jpg	PaintedIron	(5312, 2988)	JPEG	RGB	4287.385442	1441.
18	imagedata07	PaintedIron_3.jpg	PaintedIron	(5312, 2988)	JPEG	RGB	335.612517	1546.
19	imagedata07	StainlessSteel_1.jpg	StainlessSteel	(5312, 2988)	JPEG	RGB	1256.493190	1838.
20	imagedata07	StainlessSteel_2.jpg	StainlessSteel	(5312, 2988)	JPEG	RGB	1159.233761	1805.
21	imagedata07	StainlessSteel_3.jpg	StainlessSteel	(5312, 2988)	JPEG	RGB	1193.991623	1814.

22 rows × 24 columns

In [28]:

```
# Delete junk records
df = df[df.Material != 'MASK']
df = df[df.Material != 'Enh']
df
```

Out[28]:

	Folder	File	Material	Size	Format	Mode	All_Bands	Su
0	imagedata07	Aluminum_1.jpg	Aluminum	(5312, 2988)	JPEG	RGB	3795.430722	1290.
1	imagedata07	Aluminum_2.jpg	Aluminum	(5312, 2988)	JPEG	RGB	4078.341415	1382.
2	imagedata07	Aluminum_3.jpg	Aluminum	(5312, 2988)	JPEG	RGB	3956.096638	1345.
3	imagedata07	Aluminum_4.jpg	Aluminum	(5312, 2988)	JPEG	RGB	3795.430722	1290.
4	imagedata07	Brass_1.jpg	Brass	(5312, 2988)	JPEG	RGB	809.915985	1720.
5	imagedata07	Brass_2.jpg	Brass	(5312, 2988)	JPEG	RGB	795.419913	1723.
6	imagedata07	Brass_3.jpg	Brass	(5312, 2988)	JPEG	RGB	808.952895	1728.
7	imagedata07	CopperWire_1.jpg	CopperWire	(5312, 2988)	JPEG	RGB	1701.825834	2024.
8	imagedata07	CopperWire_2.jpg	CopperWire	(5312, 2988)	JPEG	RGB	1631.193961	2008.
9	imagedata07	CopperWire_3.jpg	CopperWire	(5312, 2988)	JPEG	RGB	1636.447924	1991.
10	imagedata07	Copper_1.jpg	Copper	(5312, 2988)	JPEG	RGB	1037.993365	1828.
11	imagedata07	Copper_2.jpg	Copper	(5312, 2988)	JPEG	RGB	1393.597871	1951.
12	imagedata07	Copper_3.jpg	Copper	(5312, 2988)	JPEG	RGB	1339.507502	1905.
13	imagedata07	Iron_1.jpg	Iron	(5312, 2988)	JPEG	RGB	290.375493	1527.
14	imagedata07	Iron_2.jpg	Iron	(5312, 2988)	JPEG	RGB	563.473852	1618.
15	imagedata07	Iron_3.jpg	Iron	(5312, 2988)	JPEG	RGB	328.077431	1535.
16	imagedata07	PaintedIron_1.jpg	PaintedIron	(5312, 2988)	JPEG	RGB	71.007131	1476.
17	imagedata07	PaintedIron_2.jpg	PaintedIron	(5312, 2988)	JPEG	RGB	4287.385442	1441.
18	imagedata07	PaintedIron_3.jpg	PaintedIron	(5312, 2988)	JPEG	RGB	335.612517	1546.
19	imagedata07	StainlessSteel_1.jpg	StainlessSteel	(5312, 2988)	JPEG	RGB	1256.493190	1838.
20	imagedata07	StainlessSteel_2.jpg	StainlessSteel	(5312, 2988)	JPEG	RGB	1159.233761	1805.
21	imagedata07	StainlessSteel_3.jpg	StainlessSteel	(5312, 2988)	JPEG	RGB	1193.991623	1814.

22 rows × 24 columns

Write statistics in excel book

In [30]:

```
# Verify my current folder
path = mypath + r"/upt_data.xlsx"
print('Write statistics into file :', path)

# Block to Read excel old excel file
book = load_workbook(path)
writer = pd.ExcelWriter(path, engine = 'openpyxl')
writer.book = book
# ------------------------

# Write statistics into excel file
#writer = pd.ExcelWriter(path, engine = 'xlsxwriter') # only for new excelfile
df.to_excel(writer, sheet_name = folder)
writer.save()
writer.close()
```

```
Write statistics into file : C:\Users\manuel.robalinho\Google Drive\UPT_Po
rtucalense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/upt_data.x
lsx
```

Plot

In [31]:

```
df_plot = pd.DataFrame(df, columns=["Material", "All_Bands", "Sum_Ch0", "Sum_Ch1", "Sum
_Ch2",
                                    "Color", "Color_RGB", "Color_hex","Color_dec",
                                    "Med_Extrems", "Max_Histog", "Idx_Max_Histog","Min_
Histog",
                                    "Idx_Min_Histog","perc_R", "perc_G", "perc_B"])
df_plot
```

Out[31]:

	Material	All_Bands	Sum_Ch0	Sum_Ch1	Sum_Ch2	Color	Color_R
0	Aluminum	3795.430722	1290.867334	1264.488693	1240.074695	azure	(244, 2 2
1	Aluminum	4078.341415	1382.075655	1355.066322	1341.199438	azure	(244, 2 2
2	Aluminum	3956.096638	1345.202894	1312.189751	1298.703993	azure	(244, 2 2
3	Aluminum	3795.430722	1290.867334	1264.488693	1240.074695	azure	(244, 2 2
4	Brass	809.915985	1720.902454	1712.269126	1671.711701	khaki	(253, 2 1
5	Brass	795.419913	1723.301362	1707.566316	1659.519531	khaki	(253, 2 1
6	Brass	808.952895	1728.201794	1713.305607	1662.412790	khaki	(253, 2 1
7	CopperWire	1701.825834	2024.420058	2000.821427	1971.551645	khaki	(253, 2 1
8	CopperWire	1631.193961	2008.148997	1975.862114	1942.150146	khaki	(253, 2 1
9	CopperWire	1636.447924	1991.258699	1982.614786	1957.541735	khaki	(253, 2 1
10	Copper	1037.993365	1828.316657	1783.668242	1720.975762	khaki	(253, 2 1
11	Copper	1393.597871	1951.851412	1897.481139	1839.232616	lemonchiffon	(253, 2 1
12	Copper	1339.507502	1905.950632	1882.785425	1845.738741	lemonchiffon	(253, 2 1
13	Iron	290.375493	1527.384117	1534.573907	1523.384765	khaki	(253, 2 1
14	Iron	563.473852	1618.699193	1631.213833	1608.528122	lemonchiffon	(253, 2 1
15	Iron	328.077431	1535.456848	1562.190463	1525.397416	khaki	(253, 2 1
16	PaintedIron	71.007131	1476.931962	1451.601158	1437.441307	azure	(244, 2 2
17	PaintedIron	4287.385442	1441.211650	1429.451259	1416.722533	azure	(244, 2 2
18	PaintedIron	335.612517	1546.023623	1548.193045	1536.363145	azure	(244, 2 2
19	StainlessSteel	1256.493190	1838.259548	1852.298943	1860.901995	azure	(244, 2 2
20	StainlessSteel	1159.233761	1805.040568	1821.808232	1827.352257	azure	(244, 2 2
21	StainlessSteel	1193.991623	1814.545387	1837.406699	1837.006833	azure	(244, 2 2

In [32]:

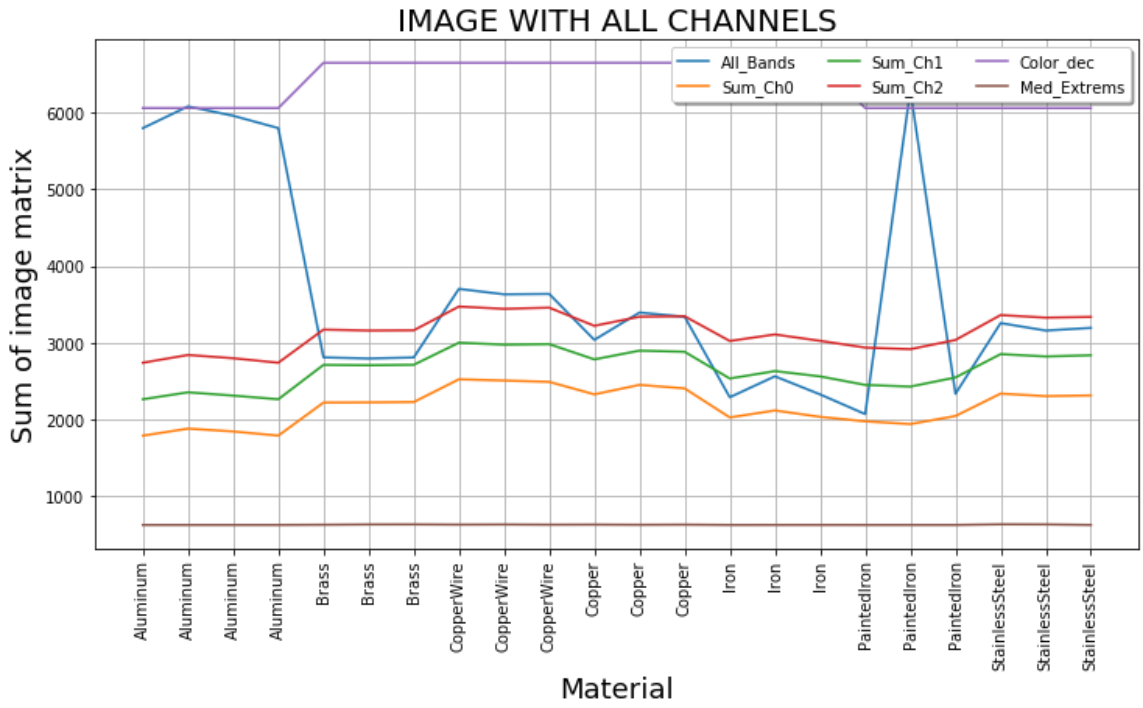
```
# Adjust values to plot
df_plot.Sum_Ch0        = df_plot.Sum_Ch0 + 500 # to have diference lines during plot
df_plot.Sum_Ch1        = df_plot.Sum_Ch1 + 1000
df_plot.Sum_Ch2        = df_plot.Sum_Ch2 + 1500
df_plot.All_Bands      = df_plot.All_Bands + 2000

df_plot.Color_dec      = df_plot.Color_dec / 1000
df_plot.Color_dec      = df_plot.Color_dec - 10000
df_plot.Med_Extrems    = df_plot.Med_Extrems + 500
df_plot.Max_Histog     = df_plot.Max_Histog / 1000
df_plot.Idx_Max_Histog = df_plot.Idx_Max_Histog + 1000
df_plot.Min_Histog     = df_plot.Min_Histog * 100
df_plot.Idx_Min_Histog = df_plot.Idx_Min_Histog * 10

df_plot.perc_R = df_plot.perc_R + 1000
df_plot.perc_G = df_plot.perc_G + 1100
df_plot.perc_B = df_plot.perc_B + 1200

df_plot
```

Out[32]:

	Material	All_Bands	Sum_Ch0	Sum_Ch1	Sum_Ch2	Color	Color_R
0	Aluminum	5795.430722	1790.867334	2264.488693	2740.074695	azure	(244, 2 2
1	Aluminum	6078.341415	1882.075655	2355.066322	2841.199438	azure	(244, 2 2
2	Aluminum	5956.096638	1845.202894	2312.189751	2798.703993	azure	(244, 2 2
3	Aluminum	5795.430722	1790.867334	2264.488693	2740.074695	azure	(244, 2 2
4	Brass	2809.915985	2220.902454	2712.269126	3171.711701	khaki	(253, 2 1
5	Brass	2795.419913	2223.301362	2707.566316	3159.519531	khaki	(253, 2 1
6	Brass	2808.952895	2228.201794	2713.305607	3162.412790	khaki	(253, 2 1
7	CopperWire	3701.825834	2524.420058	3000.821427	3471.551645	khaki	(253, 2 1
8	CopperWire	3631.193961	2508.148997	2975.862114	3442.150146	khaki	(253, 2 1
9	CopperWire	3636.447924	2491.258699	2982.614786	3457.541735	khaki	(253, 2 1
10	Copper	3037.993365	2328.316657	2783.668242	3220.975762	khaki	(253, 2 1
11	Copper	3393.597871	2451.851412	2897.481139	3339.232616	lemonchiffon	(253, 2 1
12	Copper	3339.507502	2405.950632	2882.785425	3345.738741	lemonchiffon	(253, 2 1
13	Iron	2290.375493	2027.384117	2534.573907	3023.384765	khaki	(253, 2 1
14	Iron	2563.473852	2118.699193	2631.213833	3108.528122	lemonchiffon	(253, 2 1
15	Iron	2328.077431	2035.456848	2562.190463	3025.397416	khaki	(253, 2 1
16	PaintedIron	2071.007131	1976.931962	2451.601158	2937.441307	azure	(244, 2 2
17	PaintedIron	6287.385442	1941.211650	2429.451259	2916.722533	azure	(244, 2 2
18	PaintedIron	2335.612517	2046.023623	2548.193045	3036.363145	azure	(244, 2 2
19	StainlessSteel	3256.493190	2338.259548	2852.298943	3360.901995	azure	(244, 2 2
20	StainlessSteel	3159.233761	2305.040568	2821.808232	3327.352257	azure	(244, 2 2
21	StainlessSteel	3193.991623	2314.545387	2837.406699	3337.006833	azure	(244, 2 2

In [33]:

```
df_plot.plot(y=["All_Bands","Sum_Ch0","Sum_Ch1", "Sum_Ch2","Color_dec","Med_Extrems"],
             figsize=(12,6), grid=True )

# Obtain legend (xticks) for X axis
loc_Array_sum = np.arange(len(df_plot.index))
# Position of X labels
xtick_loc = list(loc_Array_sum)
# Name of x labels
xticks = list(df_plot.Material)
#-------

#plt.plot(df_plot.Array_sum)
plt.title('IMAGE WITH ALL CHANNELS',fontsize=20)
plt.ylabel('Sum of image matrix',fontsize=18)
plt.xticks(xtick_loc, df_plot.Material, rotation=90)
plt.xlabel('Material',fontsize=18)
plt.legend(loc='upper right', ncol=3, fancybox=True, shadow=True)
plt.savefig(folder+"_Line Graph all channels information.png")
plt.show()
```

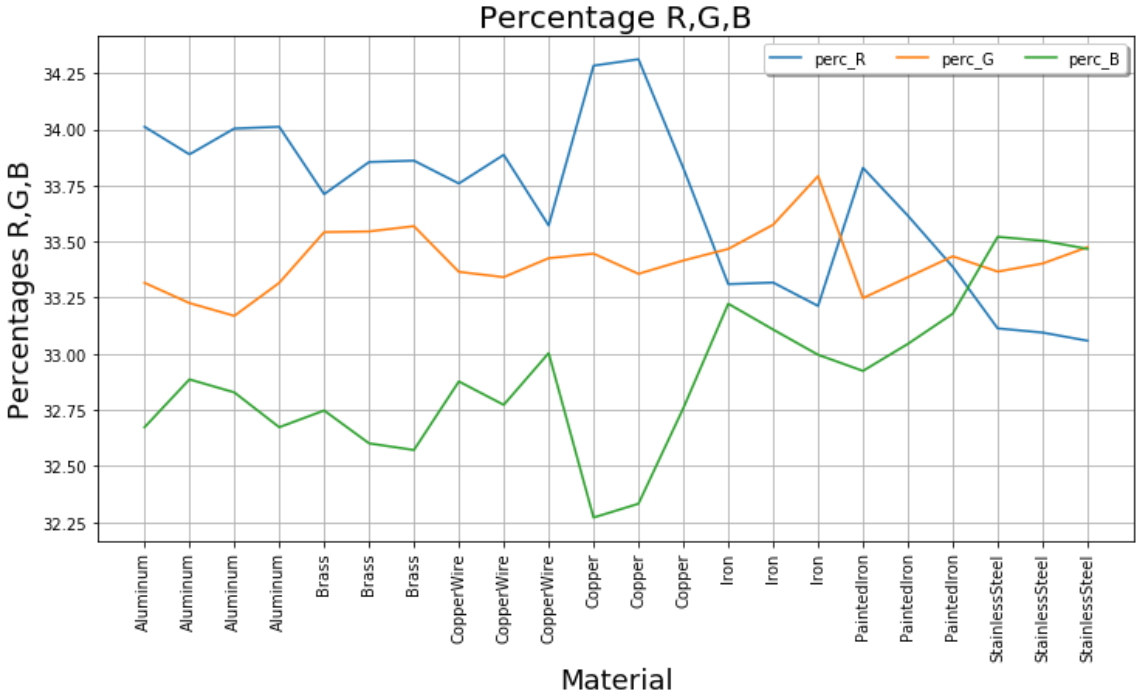


In [34]:

```
df_plot.perc_R = df_plot.perc_R - 1000
df_plot.perc_G = df_plot.perc_G - 1100
df_plot.perc_B = df_plot.perc_B - 1200
```

In [35]:

```
df_plot.plot(y=["perc_R","perc_G","perc_B"],
             figsize=(12,6), grid=True )

# Obtain legend (xticks) for X axis
loc_Array_sum = np.arange(len(df_plot.index))
# Position of X labels
xtick_loc = list(loc_Array_sum)
# Name of x labels
xticks = list(df_plot.Material)
#-------

#plt.plot(df_plot.Array_sum)
plt.title('Percentage R,G,B',fontsize=20)
plt.ylabel('Percentages R,G,B',fontsize=18)
plt.xticks(xtick_loc, df_plot.Material, rotation=90)
plt.xlabel('Material',fontsize=18)
plt.legend(loc='upper right', ncol=3, fancybox=True, shadow=True)
plt.savefig(folder+"_Line Graph Percentage RGB.png")
plt.show()
```



In [36]:

```
# Create pivot table
df_plot1 = df_plot.groupby('Material')['All_Bands', 'Sum_Ch0','Sum_Ch1','Sum_Ch2','Colo
r_dec',
                                       'Med_Extrems', 'Max_Histog', 'Idx_Max_Histog','M
in_Histog',
                                       'Idx_Min_Histog','perc_R','perc_G','perc_B'].mea
n()
df_plot1
```

Out[36]:

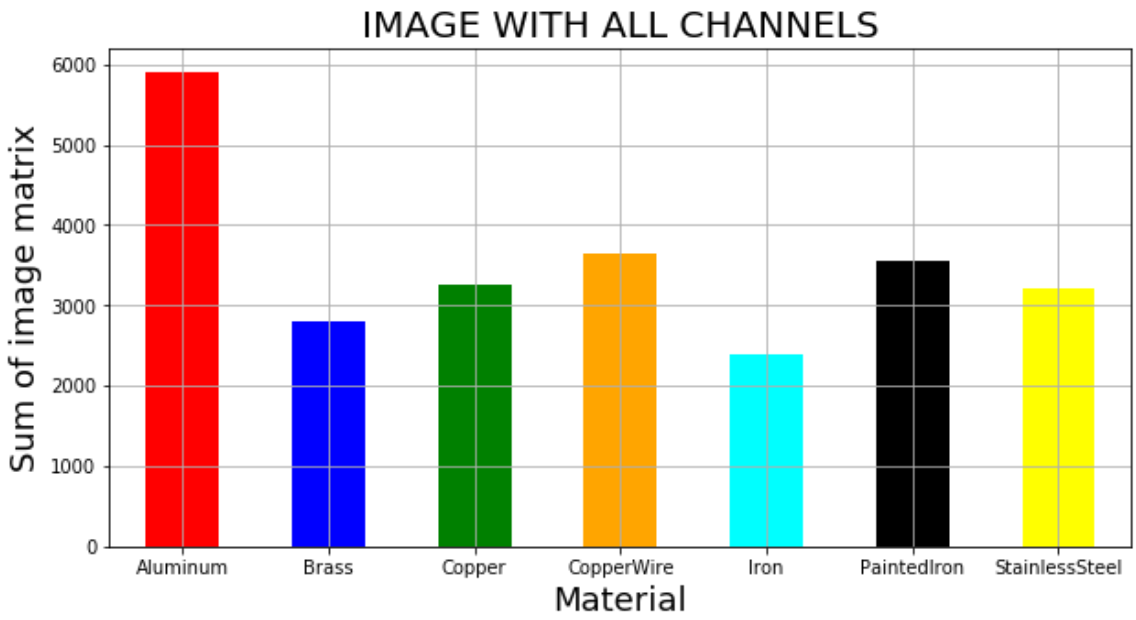
Material	All_Bands	Sum_Ch0	Sum_Ch1	Sum_Ch2	Color_dec	Med_Extren
Aluminum	5906.324874	1827.253304	2299.058365	2780.013205	6056.319000	627.5000
Brass	2804.762931	2224.135203	2711.047016	3164.548007	6646.015000	633.0555
Copper	3257.032913	2395.372900	2854.644935	3301.982373	6646.057667	631.2222
CopperWire	3656.489240	2507.942585	2986.432776	3457.081175	6646.015000	632.2777
Iron	2393.975592	2060.513386	2575.992734	3052.436768	6646.036333	627.6666
PaintedIron	3564.668363	1988.055745	2476.415154	2963.508995	6056.319000	627.5000
StainlessSteel	3203.239525	2319.281834	2837.171291	3341.753695	6056.319000	632.7777

In [37]:

```
color = ['red','blue','green','orange','cyan','black','yellow']
```

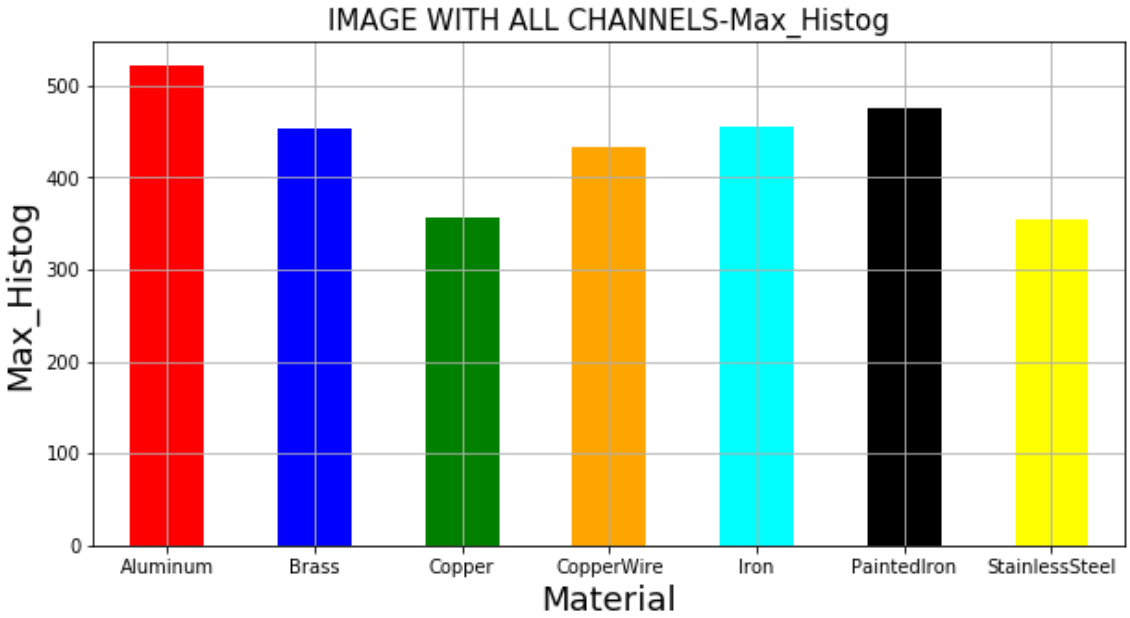
In [38]:

```
df_All_Bands = pd.DataFrame(df_plot1.All_Bands)

df_All_Bands.plot(kind='bar', y=0, color=color, legend=False, rot=0, figsize=(10,5))
plt.title('IMAGE WITH ALL CHANNELS',fontsize=20)
plt.grid(True)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Sum of image matrix',fontsize=18)
plt.savefig(folder+"_Sum of image matrix.png")
plt.show()
```



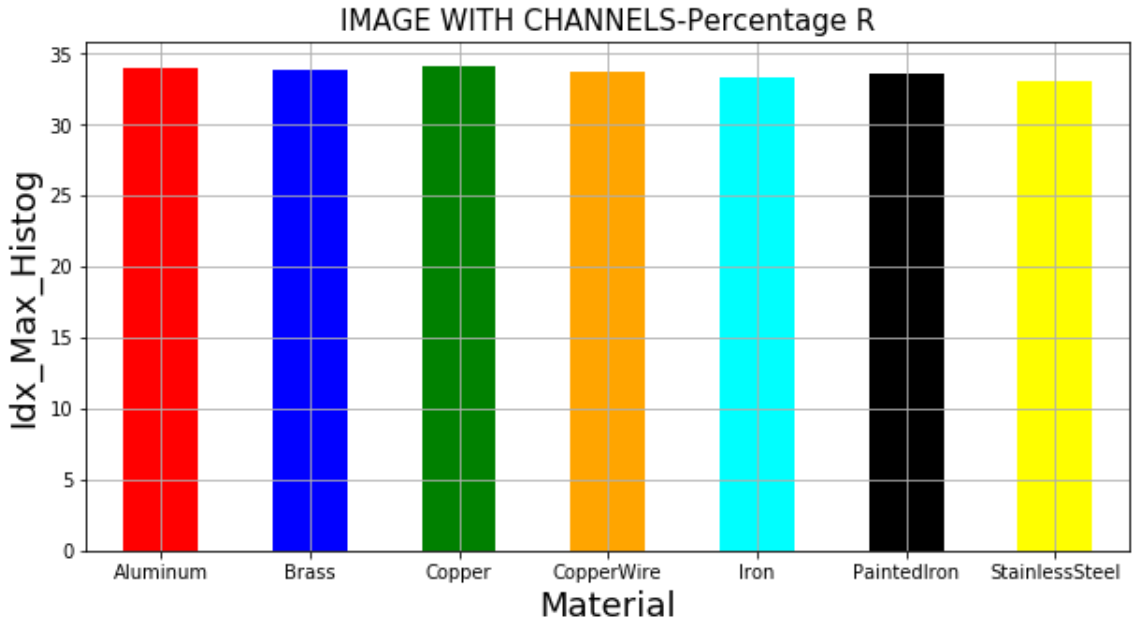
In [39]:

```
df_Max_Histog = pd.DataFrame(df_plot1.Max_Histog)

df_Max_Histog.plot(kind='bar', y=0, color=color, legend=False, rot=0, figsize=(10,5))
plt.title('IMAGE WITH ALL CHANNELS-Max_Histog',fontsize=15)
plt.grid(True)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Max_Histog',fontsize=18)
plt.savefig(folder+"_Max_Histog.png")
plt.show()
```



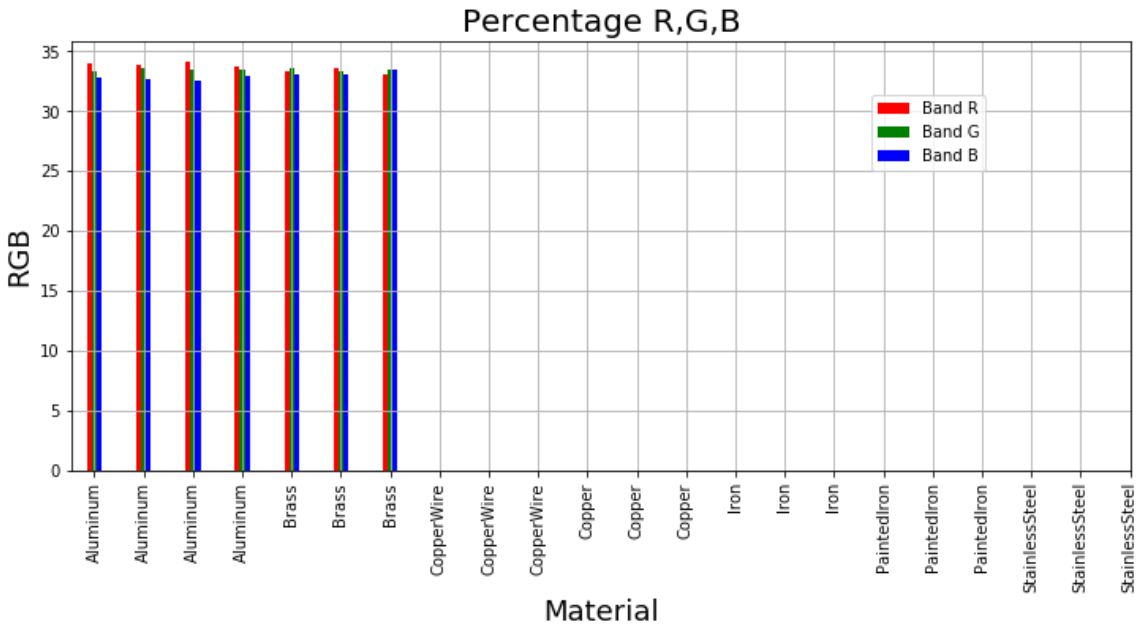
In [43]:

```
df_perc = pd.DataFrame(df_plot1.perc_R)

df_perc.plot(kind='bar', y=0, color=color, legend=False, rot=0, figsize=(10,5))
plt.title('IMAGE WITH CHANNELS-Percentage R',fontsize=15)
plt.grid(True)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Idx_Max_Histog',fontsize=18)
plt.show()
```



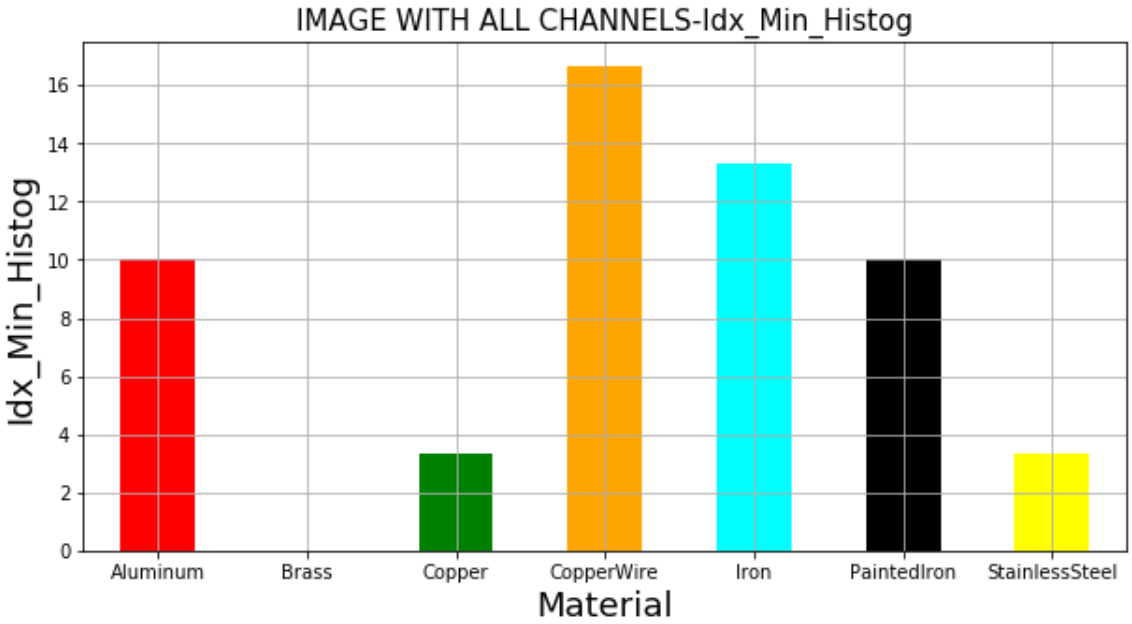
In [44]:

```
loc_Array_sum = np.arange(len(df_plot1.index))+0.1 # Offsetting the tick-label location

loc_r = np.arange(len(df_plot1.index))-0.1 # Offsetting the tick-label location
loc_g = np.arange(len(df_plot1.index))-0.0 # Offsetting the tick-label location
loc_b = np.arange(len(df_plot1.index))+0.1 # Offsetting the tick-label location

#xtick_loc = list(loc_Array_sum) + list(loc_r) + list(loc_g) + list(loc_b)
#xticks = list(selected.keys())+ list(rejected.keys())
colors = ['darkred','red','green','blue','orange','cyan','black','yellow']
plt.figure(figsize=(12,5))

plt.bar(loc_r, df_plot1.perc_R, color='red', width=0.1, label='Band R')
plt.bar(loc_g, df_plot1.perc_G, color='green', width=0.1,label='Band G')
plt.bar(loc_b, df_plot1.perc_B, color='blue', width=0.1,label='Band B')

plt.title('Percentage R,G,B',fontsize=20)
plt.xlabel('Material',fontsize=18)
plt.ylabel('RGB',fontsize=18)
plt.grid(True)
plt.xticks(xtick_loc, xticks, rotation=90)
plt.legend(bbox_to_anchor=(.8,0.8),\
    bbox_transform=plt.gcf().transFigure)
plt.savefig(folder+"_Bar Diagram_perc_RGB.png")
plt.show()
```



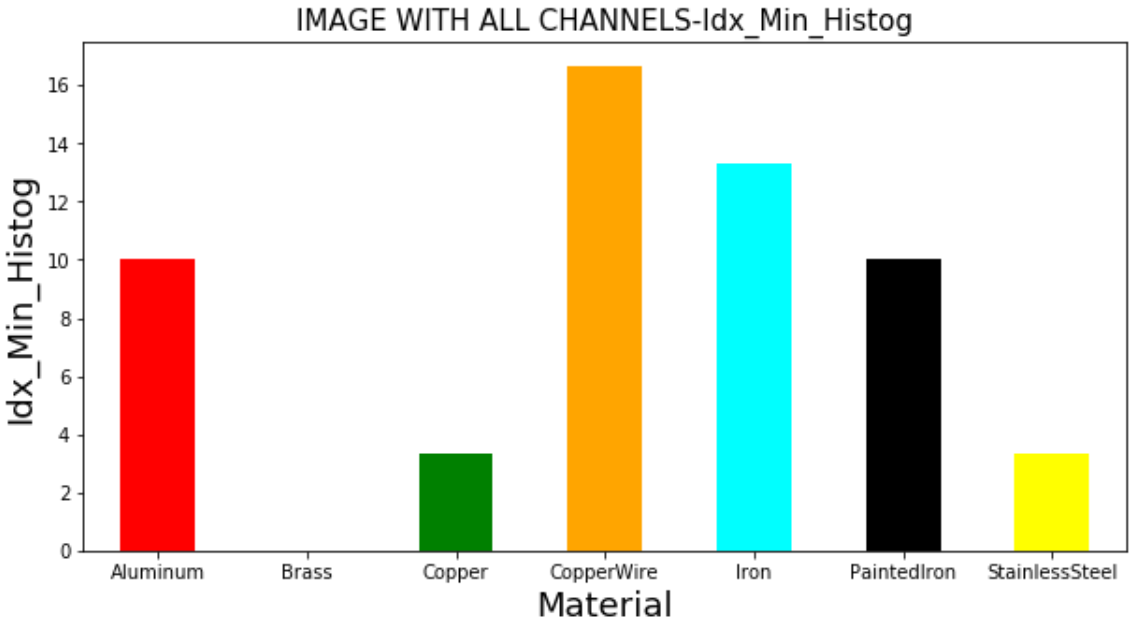
In [45]:

```
df_Idx_Min_Histog = pd.DataFrame(df_plot1.Idx_Min_Histog)

df_Idx_Min_Histog.plot(kind='bar', y=0, color=color, legend=False, rot=0, figsize=(10,5
))
plt.title('IMAGE WITH ALL CHANNELS-Idx_Min_Histog',fontsize=15)
plt.grid(True)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Idx_Min_Histog',fontsize=18)
plt.savefig(folder+"_Idx_Min_Histogram.png")
plt.show()
```



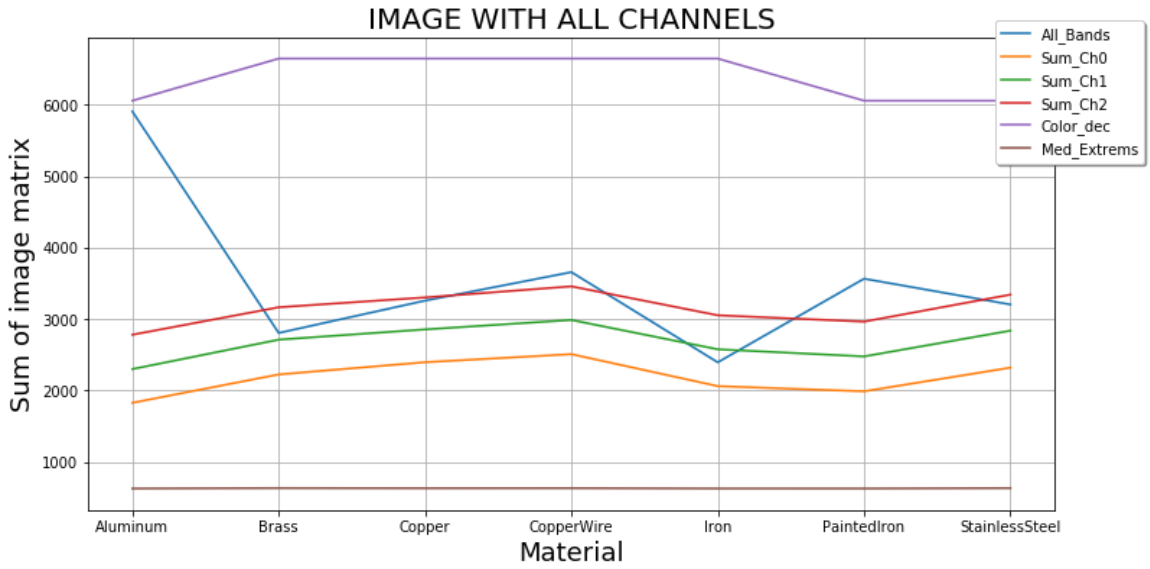
In [46]:

```
df_Idx_Min_Histog = pd.DataFrame(df_plot1.Idx_Min_Histog)

df_Idx_Min_Histog.plot(kind='bar', y=0, color=color, legend=False, rot=0, figsize=(10,5
))
plt.title('IMAGE WITH ALL CHANNELS-Idx_Min_Histog',fontsize=15)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Idx_Min_Histog',fontsize=18)
plt.show()
```



In [47]:

```
loc_Array_sum = np.arange(len(df_plot1.index))
xtick_loc = list(loc_Array_sum)
xticks = list(df_plot1.index)

df_plot1.plot( y=["All_Bands","Sum_Ch0","Sum_Ch1", "Sum_Ch2","Color_dec","Med_Extrems"
],
              figsize=(12,6), grid=True )
plt.xticks(xtick_loc, df_plot1.index, rotation=0)
plt.title('IMAGE WITH ALL CHANNELS',fontsize=20)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Sum of image matrix',fontsize=18)
plt.legend(loc='upper right', ncol=1, fancybox=True, shadow=True,bbox_to_anchor=(1.1,
1.05))
plt.savefig(folder+"_resume all channels.png")
plt.show()
```



In [48]:

```
df_plot1
```

Out[48]:

Material	All_Bands	Sum_Ch0	Sum_Ch1	Sum_Ch2	Color_dec	Med_Extren
Aluminum	5906.324874	1827.253304	2299.058365	2780.013205	6056.319000	627.5000
Brass	2804.762931	2224.135203	2711.047016	3164.548007	6646.015000	633.0555
Copper	3257.032913	2395.372900	2854.644935	3301.982373	6646.057667	631.2222
CopperWire	3656.489240	2507.942585	2986.432776	3457.081175	6646.015000	632.2777
Iron	2393.975592	2060.513386	2575.992734	3052.436768	6646.036333	627.6666
PaintedIron	3564.668363	1988.055745	2476.415154	2963.508995	6056.319000	627.5000
StainlessSteel	3203.239525	2319.281834	2837.171291	3341.753695	6056.319000	632.7777

In [49]:

```
# Copy dataframe to arrange values
df_plot2 =  df_plot1.copy()
df_plot2
```

Out[49]:

Material	All_Bands	Sum_Ch0	Sum_Ch1	Sum_Ch2	Color_dec	Med_Extren
Aluminum	5906.324874	1827.253304	2299.058365	2780.013205	6056.319000	627.5000
Brass	2804.762931	2224.135203	2711.047016	3164.548007	6646.015000	633.0555
Copper	3257.032913	2395.372900	2854.644935	3301.982373	6646.057667	631.2222
CopperWire	3656.489240	2507.942585	2986.432776	3457.081175	6646.015000	632.2777
Iron	2393.975592	2060.513386	2575.992734	3052.436768	6646.036333	627.6666
PaintedIron	3564.668363	1988.055745	2476.415154	2963.508995	6056.319000	627.5000
StainlessSteel	3203.239525	2319.281834	2837.171291	3341.753695	6056.319000	632.7777

In [50]:

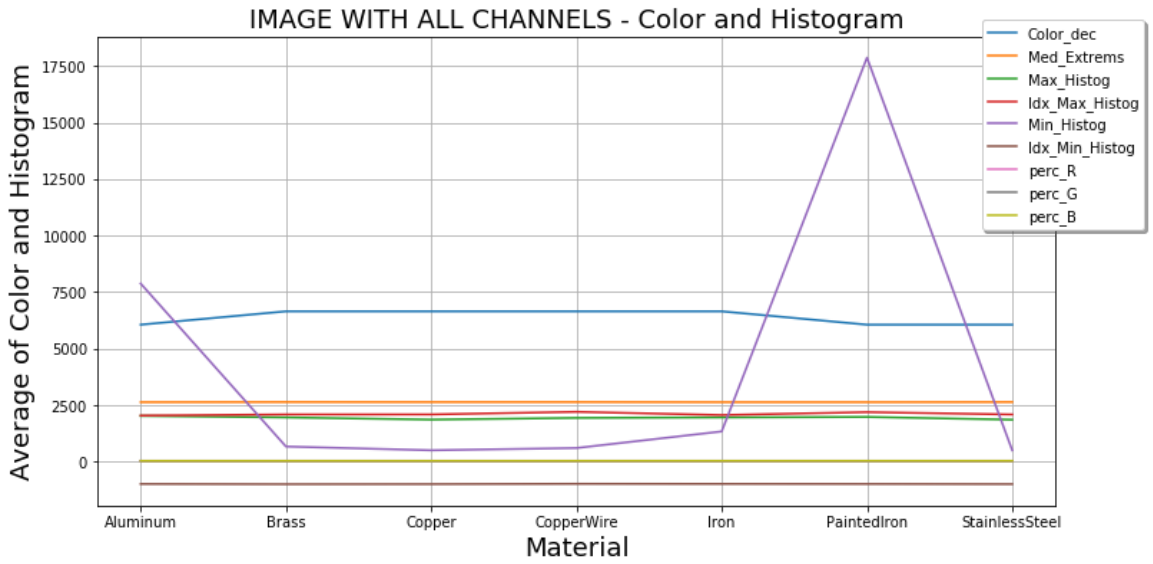
```
df_plot2.Med_Extrems    = df_plot2.Med_Extrems + 2000
df_plot2.Max_Histog     = df_plot2.Max_Histog  + 1500
df_plot2.Idx_Max_Histog = df_plot2.Idx_Max_Histog + 1000
df_plot2.Min_Histog     = df_plot2.Min_Histog + 500
df_plot2.Idx_Min_Histog = df_plot2.Idx_Min_Histog - 1000
df_plot2.head()
```

Out[50]:

Material	All_Bands	Sum_Ch0	Sum_Ch1	Sum_Ch2	Color_dec	Med_Extrems
Aluminum	5906.324874	1827.253304	2299.058365	2780.013205	6056.319000	2627.500000
Brass	2804.762931	2224.135203	2711.047016	3164.548007	6646.015000	2633.055556
Copper	3257.032913	2395.372900	2854.644935	3301.982373	6646.057667	2631.222222
CopperWire	3656.489240	2507.942585	2986.432776	3457.081175	6646.015000	2632.277778
Iron	2393.975592	2060.513386	2575.992734	3052.436768	6646.036333	2627.666667

In [51]:

```
loc_Array_sum = np.arange(len(df_plot2.index))
xtick_loc = list(loc_Array_sum)
xticks = list(df_plot1.index)

df_plot2.plot( y=['Color_dec','Med_Extrems', 'Max_Histog', 'Idx_Max_Histog',
                  'Min_Histog','Idx_Min_Histog','perc_R','perc_G','perc_B'],figsize=(12
,6), grid=True )
plt.xticks(xtick_loc, df_plot2.index, rotation=0)
plt.title('IMAGE WITH ALL CHANNELS - Color and Histogram',fontsize=18)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Average of Color and Histogram',fontsize=18)
plt.legend(loc='upper right', ncol=1, fancybox=True, shadow=True,bbox_to_anchor=(1.1,
1.05))
plt.savefig(folder+"_color_and_histogram.png")
plt.show()
```



In [52]:

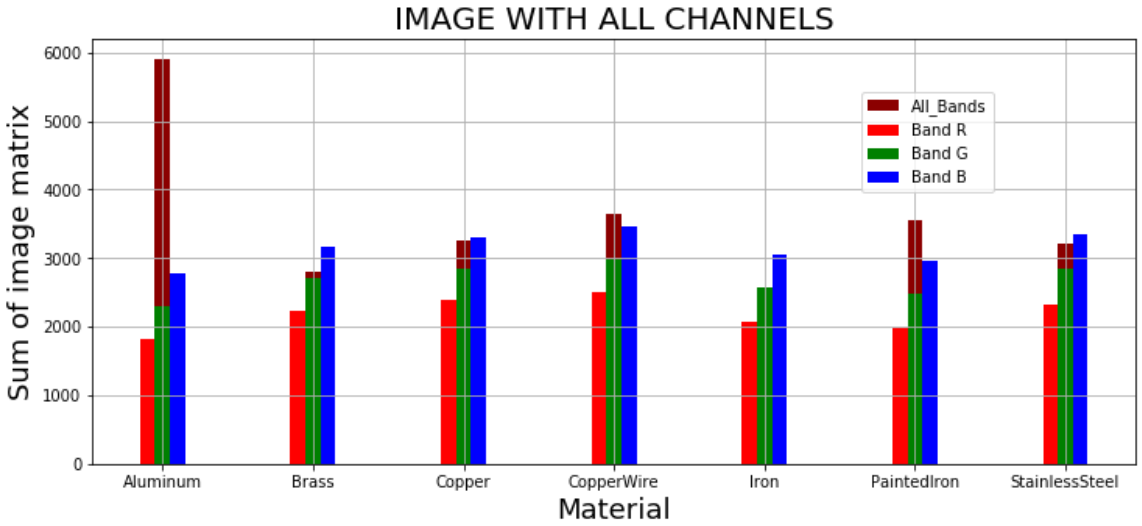
```
# Create Xlabels
loc_Array_sum = np.arange(len(df_plot1.index))+0.0 # Offsetting the tick-label location
loc_r = np.arange(len(df_plot1.index))+0.1 # Offsetting the tick-label location
loc_g = np.arange(len(df_plot1.index))-0.0 # Offsetting the tick-label location
loc_b = np.arange(len(df_plot1.index))-0.1 # Offsetting the tick-label location

xtick_loc = list(loc_g)
xticks = list(df_plot1.index)
```

In [53]:

```
# Plot
```

In [54]:

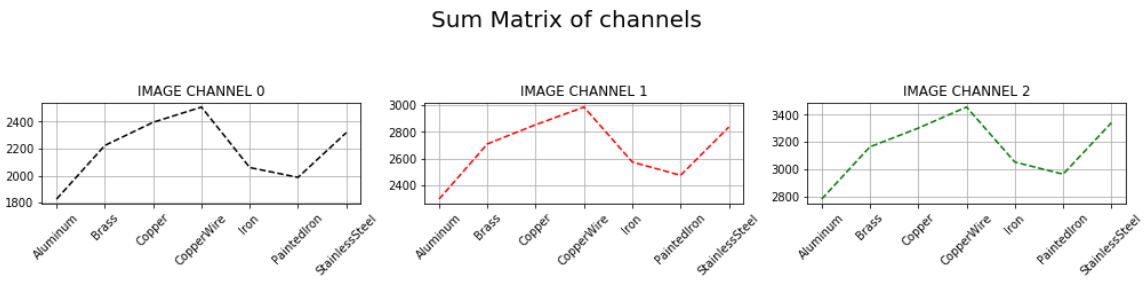
```
# Plot  Bar Graph
#df_plot1.plot(kind='bar', figsize=(12,5), grid=True, color='darkred',fontsize=18)
loc_Array_sum = np.arange(len(df_plot1.index))+0.0 # Offsetting the tick-label location
loc_b = np.arange(len(df_plot1.index))+0.1 # Offsetting the tick-label location
loc_g = np.arange(len(df_plot1.index))-0.0 # Offsetting the tick-label location
loc_r = np.arange(len(df_plot1.index))-0.1 # Offsetting the tick-label location

#xtick_loc = list(loc_Array_sum) + list(loc_r) + list(loc_g) + list(loc_b)
#xticks = list(selected.keys())+ list(rejected.keys())
colors = ['darkred','red','green','blue','orange','cyan','black','yellow']
plt.figure(figsize=(12,5))

plt.bar(loc_Array_sum, df_plot1.All_Bands, color=colors[0], width=0.1, label='All_Band
s')
plt.bar(loc_r, df_plot1.Sum_Ch0, color=colors[1], width=0.1,label='Band R')
plt.bar(loc_g, df_plot1.Sum_Ch1, color=colors[2], width=0.1,label='Band G')
plt.bar(loc_b, df_plot1.Sum_Ch2, color=colors[3], width=0.1,label='Band B')

plt.title('IMAGE WITH ALL CHANNELS',fontsize=20)
plt.grid(True)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Sum of image matrix',fontsize=18)
plt.xticks(xtick_loc, xticks, rotation=0)
plt.legend(bbox_to_anchor=(.8,0.8),\
    bbox_transform=plt.gcf().transFigure)
plt.savefig(folder+"_all bands.png")
plt.show()
```



In [55]:

```
plt.figure(1)
plt.figure(figsize=(17, 4))
plt.tight_layout()
plt.subplot(231)
plt.title('IMAGE CHANNEL 0')
plt.xticks(rotation=45)
plt.grid(True)
plt.plot(df_plot1.Sum_Ch0, 'k--')

plt.subplot(232)
plt.title('IMAGE CHANNEL 1')
plt.xticks(rotation=45)
plt.grid(True)
plt.plot(df_plot1.Sum_Ch1,  'r--')

plt.subplot(233)
plt.title('IMAGE CHANNEL 2')
plt.xticks(rotation=45)
plt.plot(df_plot1.Sum_Ch2,  'g--')
plt.grid(True)
plt.suptitle('Sum Matrix of channels',fontsize=20,y=1.08)
#plt.tight_layout()
plt.subplots_adjust(top=0.8)
plt.savefig(folder+"_Sum Matrix of channels.png")
plt.show()
```

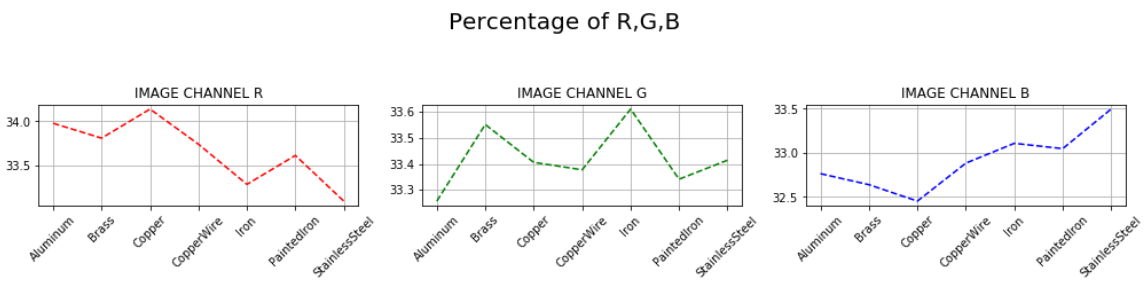
```
<Figure size 432x288 with 0 Axes>
```



In [56]:

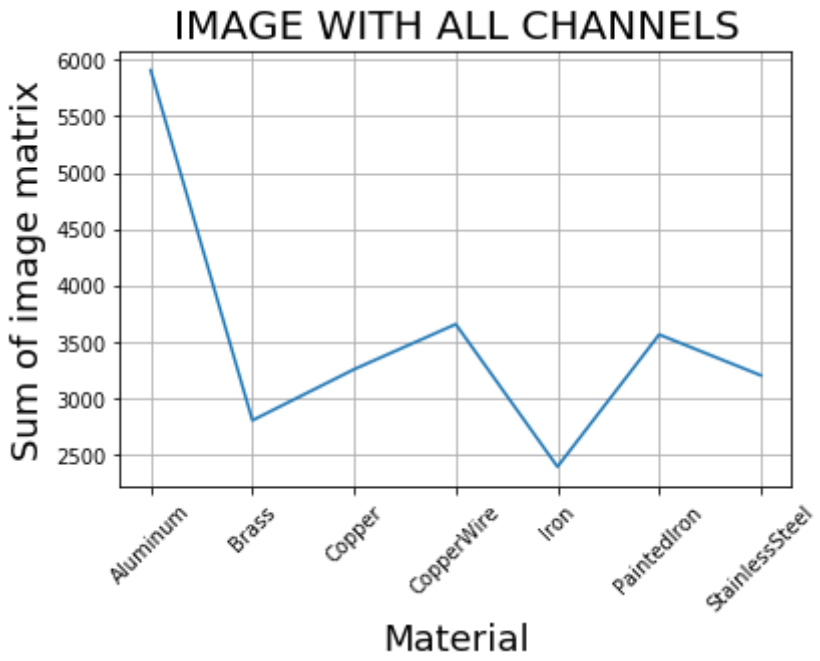
```
# Pèrcentage of R,G,B
plt.figure(1)
plt.figure(figsize=(17, 4))
plt.tight_layout()
plt.subplot(231)
plt.title('IMAGE CHANNEL R')
plt.xticks(rotation=45)
plt.grid(True)
plt.plot(df_plot1.perc_R, 'r--')

plt.subplot(232)
plt.title('IMAGE CHANNEL G')
plt.xticks(rotation=45)
plt.grid(True)
plt.plot(df_plot1.perc_G,  'g--')

plt.subplot(233)
plt.title('IMAGE CHANNEL B')
plt.xticks(rotation=45)
plt.plot(df_plot1.perc_B,  'b--')
plt.grid(True)

plt.suptitle('Percentage of R,G,B',fontsize=20,y=1.08)
#plt.tight_layout()
plt.subplots_adjust(top=0.8)
plt.savefig(folder+'_Percentage_RGB.png', bbox_inches='tight', pad_inches=0.0)
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



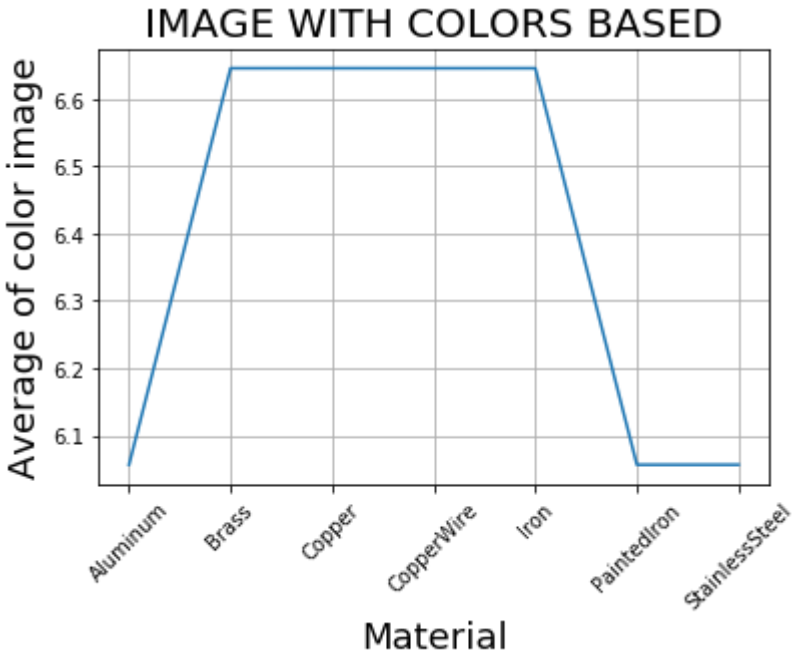
In [57]:

```
# Plot channel based
plt.plot(df_plot1.All_Bands)
plt.title('IMAGE WITH ALL CHANNELS',fontsize=20)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Sum of image matrix',fontsize=18)
plt.xticks(rotation=45)
plt.grid(True)
plt.savefig(folder+'_Sum_all_channels.png', bbox_inches='tight', pad_inches=0.0)
plt.show()
```



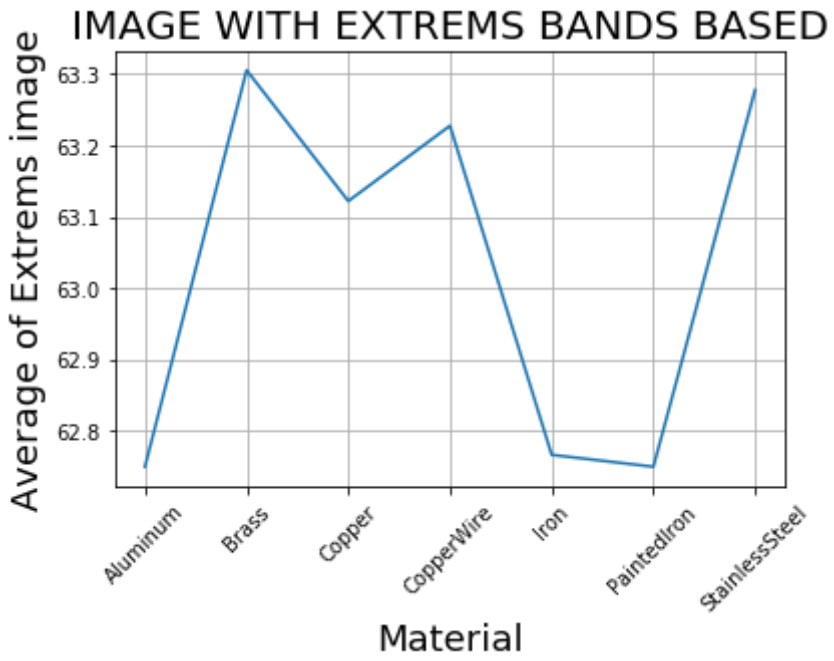
In [58]:

```
# Plot based on color
plt.plot(df_plot1.Color_dec/1000)
plt.title('IMAGE WITH COLORS BASED',fontsize=20)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Average of color image',fontsize=18)
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



In [59]:

```
# Plot based on Extrems of the Bands
plt.plot(df_plot1.Med_Extrems/10)
plt.title('IMAGE WITH EXTREMS BANDS BASED',fontsize=20)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Average of Extrems image',fontsize=18)
plt.xticks(rotation=45)
plt.grid(True)
plt.savefig(folder+'_color_based.png', bbox_inches='tight', pad_inches=0.0)
plt.show()
```



Create Histograms

https://www.cambridgeincolour.com/pt-br/tutoriais/histograms1.htm (https://www.cambridgeincolour.com/pt-br/tutoriais/histograms1.htm)

https://www.cambridgeincolour.com/pt-br/tutoriais/image-noise.htm (https://www.cambridgeincolour.com/pt-br/tutoriais/image-noise.htm)

http://www2.ic.uff.br/~aconci/aula-2-2015-AI.pdf (http://www2.ic.uff.br/~aconci/aula-2-2015-AI.pdf)

https://www.ic.unicamp.br/~ra144681/misc/files/ApostilaProcDeImagesParteI.pdf (https://www.ic.unicamp.br/~ra144681/misc/files/ApostilaProcDeImagesParteI.pdf)

histograma, também conhecido como distribuição de frequências ou diagrama das frequências, é a representação gráfica, em colunas (retângulos), de um conjunto de dados previamente tabulado e dividido em classes uniformes.

Histogramas:

O histograma de uma imagem cinza e uma funcao discreta h(I) (vetor) que produz o numero de ocorrencias de cada nıvel de cinza na imagem. O histograma normalizado h(I)/|DI | representa a distribuicao de probabilidade dos valores dos pixels.

Imagens claras possuem histogramas com altas concentracoes de pixels de alto brilho. Imagens escuras possuem histogramas com altas concentracoes de pixels de baixo brilho. O contraste maior esta associado a um grau maior de dispersao do histograma.

No caso de imagens multiespectrais, cada banda e´ requantizada em um certo numero de intervalos, de forma que o espaco de caracterısticas Zk ´e dividido em hipercubos (bins do histograma). A contagem de cores em cada bin ´e usada no c´alculo do histograma. Assim, para cada bin, precisamos analisar os n´ıveis de cinza das 3 bandas da imagem colorida (RGB).

Entendendo Histogramas:

O histograma mostra a frequencia dos valores de brilho da imagem, ou seja, a quantidade de luz presente na imagem.

In [60]:

```
list_of_images
```

Out[60]:

```
['Aluminum_1.jpg',
 'Aluminum_2.jpg',
 'Aluminum_3.jpg',
 'Aluminum_4.jpg',
 'Brass_1.jpg',
 'Brass_2.jpg',
 'Brass_3.jpg',
 'CopperWire_1.jpg',
 'CopperWire_2.jpg',
 'CopperWire_3.jpg',
 'Copper_1.jpg',
 'Copper_2.jpg',
 'Copper_3.jpg',
 'Iron_1.jpg',
 'Iron_2.jpg',
 'Iron_3.jpg',
 'PaintedIron_1.jpg',
 'PaintedIron_2.jpg',
 'PaintedIron_3.jpg',
 'StainlessSteel_1.jpg',
 'StainlessSteel_2.jpg',
 'StainlessSteel_3.jpg']
```

In [61]:

```
# Delete values from list - Bad image names
def remove_values_from_list(list_values, mask):
    list_new = list()
    for list_value in list_values:
        if(mask not in list_value):
            print(list_value)
            list_new.append(list_value)
    return list_new
```

In [62]:

```
# Remove from list names with 'MASK'
new_list = remove_values_from_list(list_of_images, 'MASK')
```

```
Aluminum_1.jpg
Aluminum_2.jpg
Aluminum_3.jpg
Aluminum_4.jpg
Brass_1.jpg
Brass_2.jpg
Brass_3.jpg
CopperWire_1.jpg
CopperWire_2.jpg
CopperWire_3.jpg
Copper_1.jpg
Copper_2.jpg
Copper_3.jpg
Iron_1.jpg
Iron_2.jpg
Iron_3.jpg
PaintedIron_1.jpg
PaintedIron_2.jpg
PaintedIron_3.jpg
StainlessSteel_1.jpg
StainlessSteel_2.jpg
StainlessSteel_3.jpg
```

In [63]:

```
# Remove from list names with 'Enh'
new_list = remove_values_from_list(new_list, 'Enh')
```

```
Aluminum_1.jpg
Aluminum_2.jpg
Aluminum_3.jpg
Aluminum_4.jpg
Brass_1.jpg
Brass_2.jpg
Brass_3.jpg
CopperWire_1.jpg
CopperWire_2.jpg
CopperWire_3.jpg
Copper_1.jpg
Copper_2.jpg
Copper_3.jpg
Iron_1.jpg
Iron_2.jpg
Iron_3.jpg
PaintedIron_1.jpg
PaintedIron_2.jpg
PaintedIron_3.jpg
StainlessSteel_1.jpg
StainlessSteel_2.jpg
StainlessSteel_3.jpg
```

In [64]:

```
list_of_images = new_list
```

In [65]:

```
path = mypath + '/' + folder + '/'
path
```

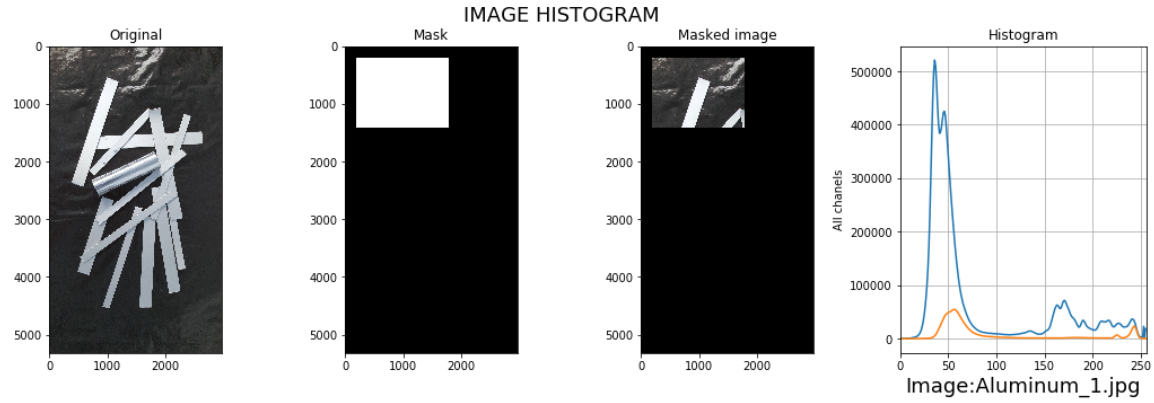
Out[65]:

```
'C:\\Users\\manuel.robalinho\\Google Drive\\UPT_Portucalense\\Trabalho fin
al\\Classificacao_Sucata\\Jupyter_Notebook/imagedata07/'
```

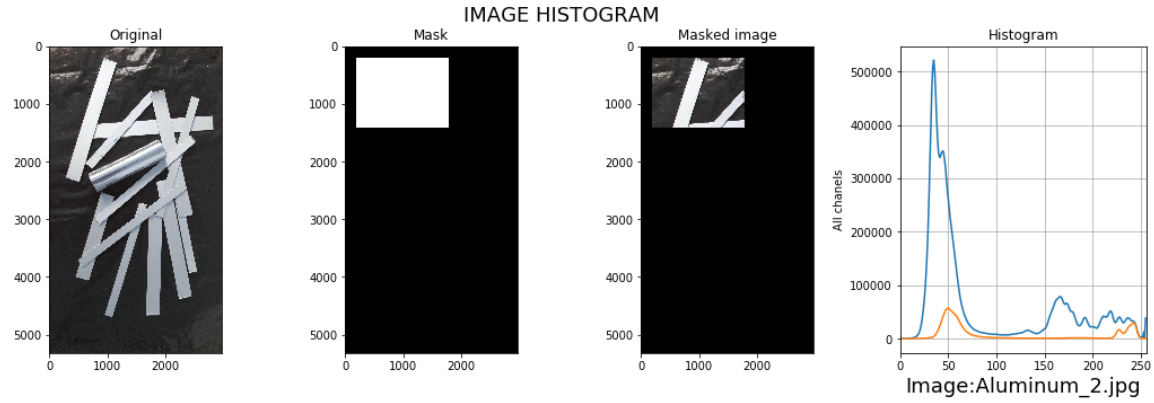
In [66]:

```
# HISTOGRAMS
# Print Histograms for all folder images
# list_of_images has all the name files

for x in list_of_images:
    print('Cv2 Histogram for File:', x)
    print_cv_hist(path, x)
```

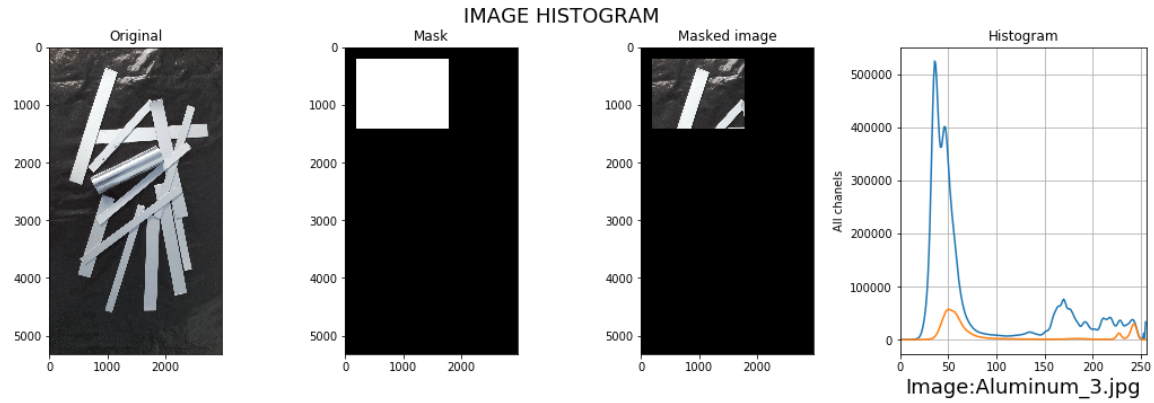
```
Cv2 Histogram for File: Aluminum_1.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Aluminu
m_1.jpg
```



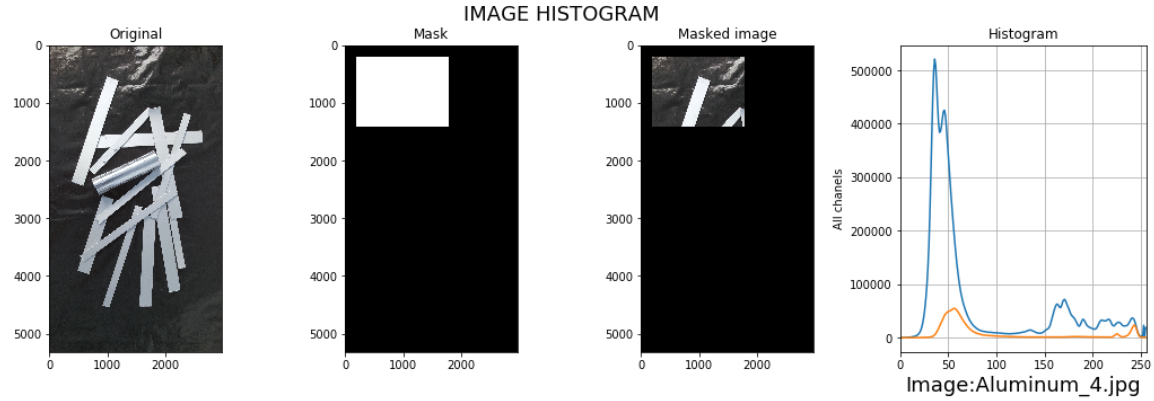

```
Cv2 Histogram for File: Aluminum_2.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Aluminu
m_2.jpg
```



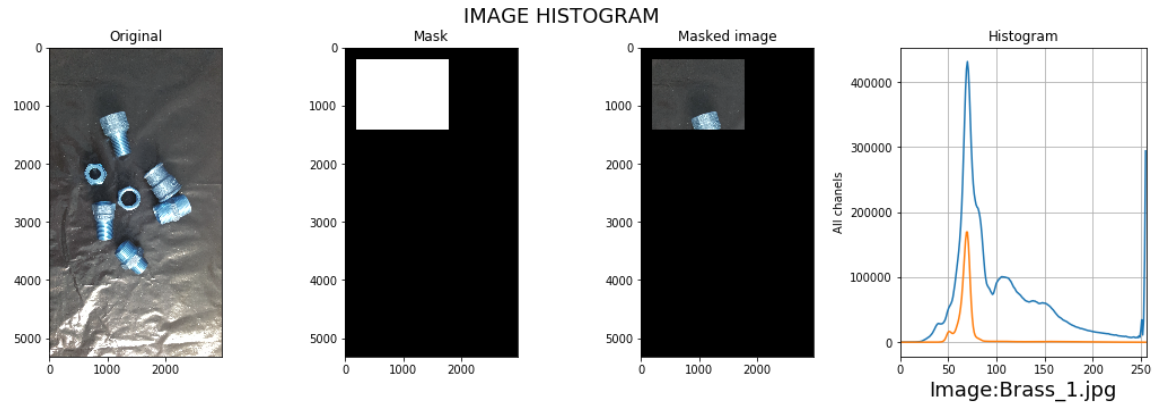

```
Cv2 Histogram for File: Aluminum_3.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Aluminu
m_3.jpg
```



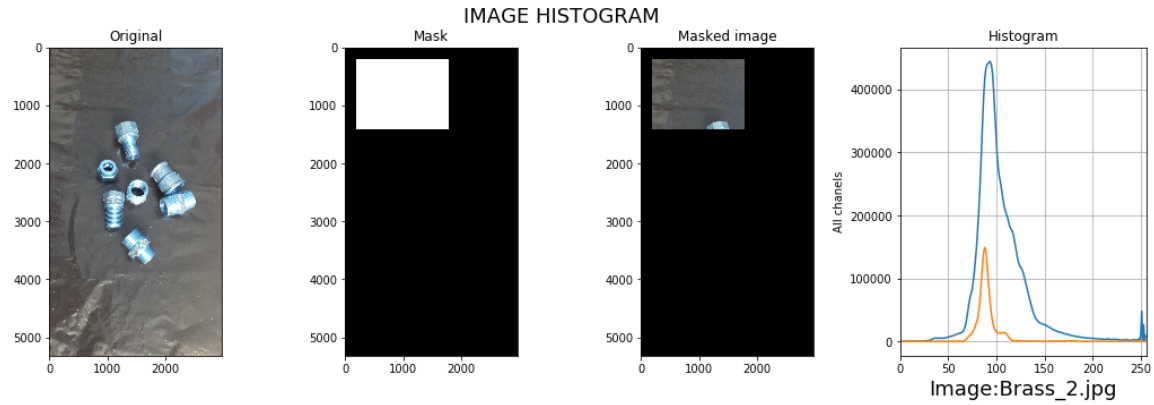

```
Cv2 Histogram for File: Aluminum_4.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Aluminu
m_4.jpg
```



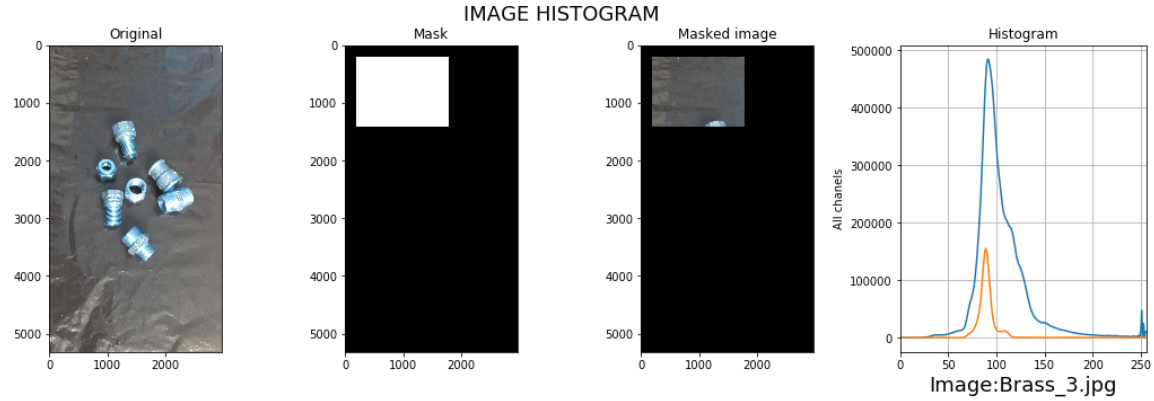

```
Cv2 Histogram for File: Brass_1.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Brass_
1.jpg
```



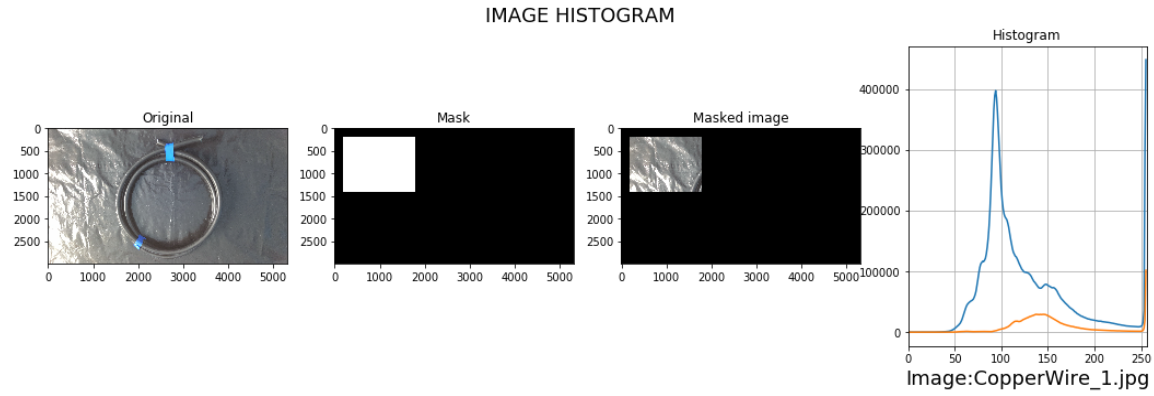

```
Cv2 Histogram for File: Brass_2.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Brass_
2.jpg
```



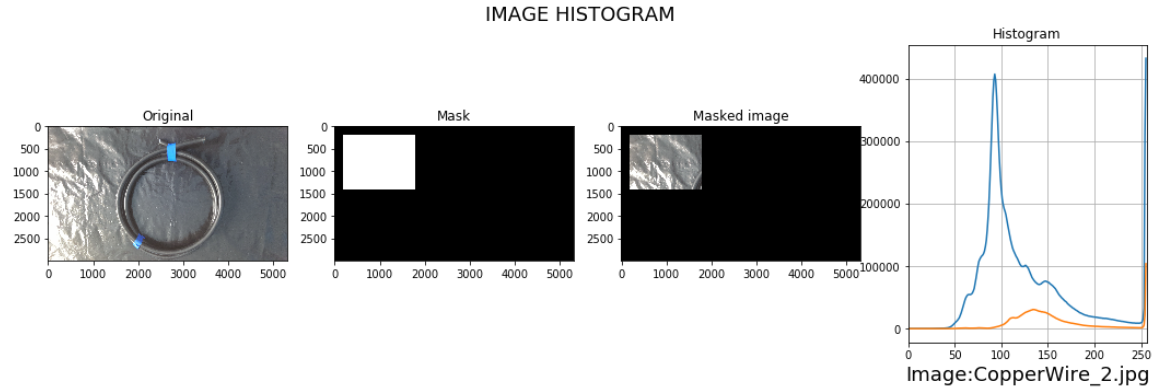

```
Cv2 Histogram for File: Brass_3.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Brass_
3.jpg
```



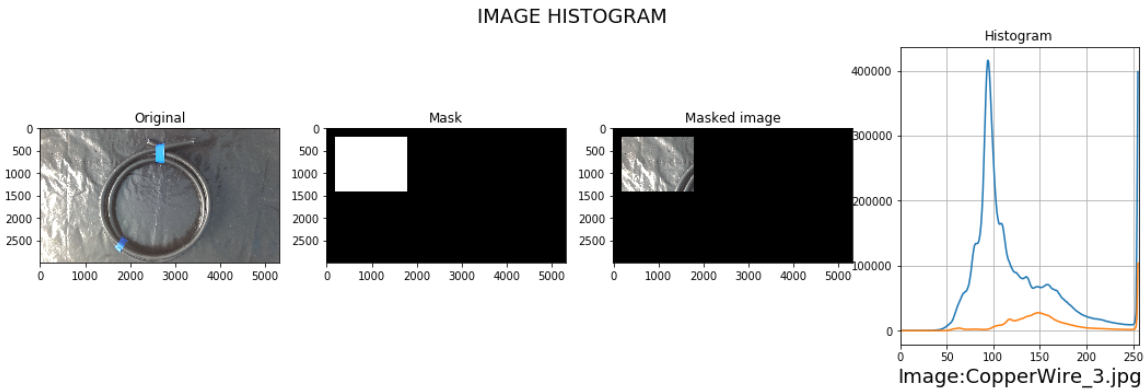

```
Cv2 Histogram for File: CopperWire_1.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/CopperW
ire_1.jpg
```



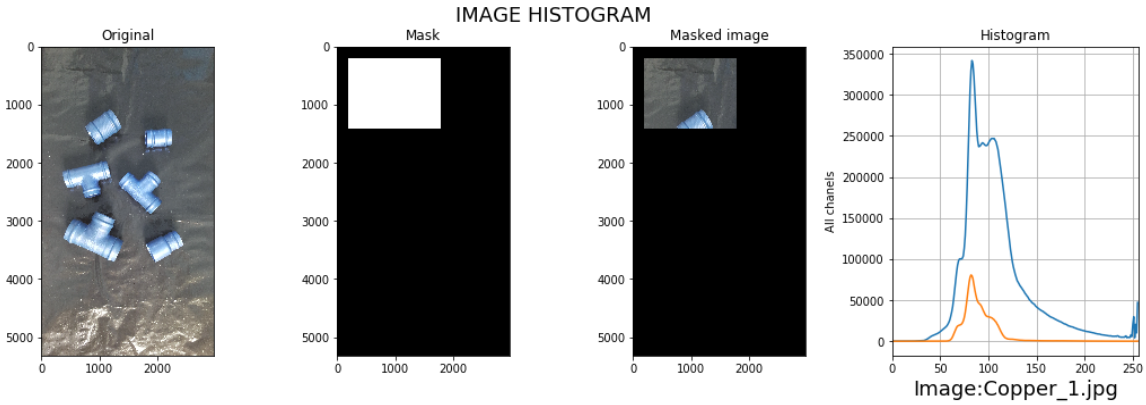

```
Cv2 Histogram for File: CopperWire_2.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/CopperW
ire_2.jpg
```



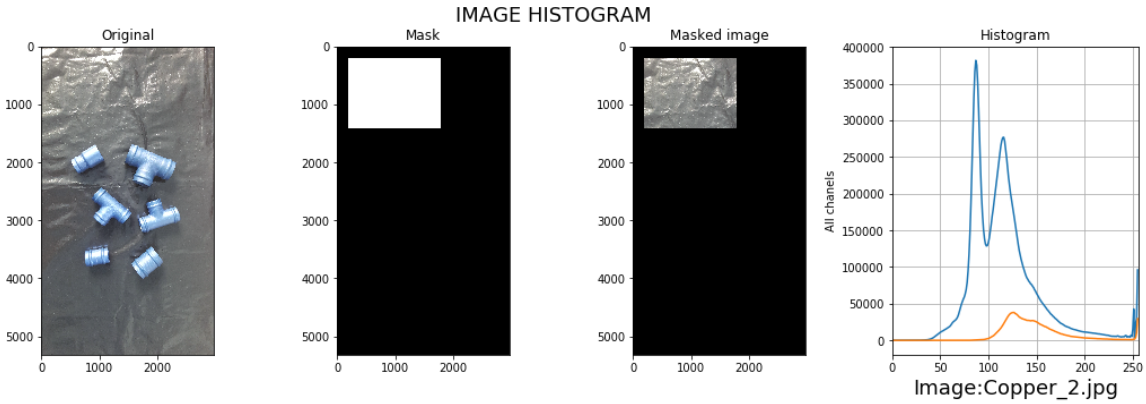

```
Cv2 Histogram for File: CopperWire_3.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/CopperW
ire_3.jpg
```



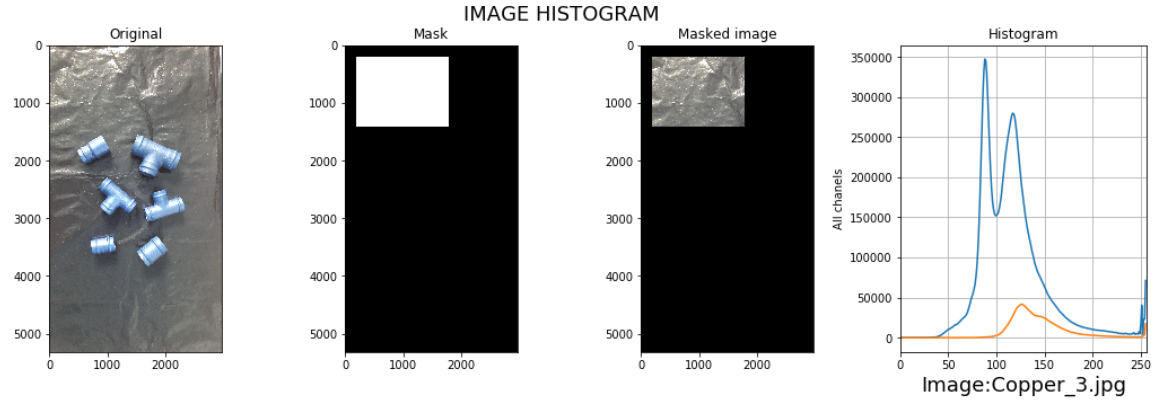

```
Cv2 Histogram for File: Copper_1.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Copper_
1.jpg
```



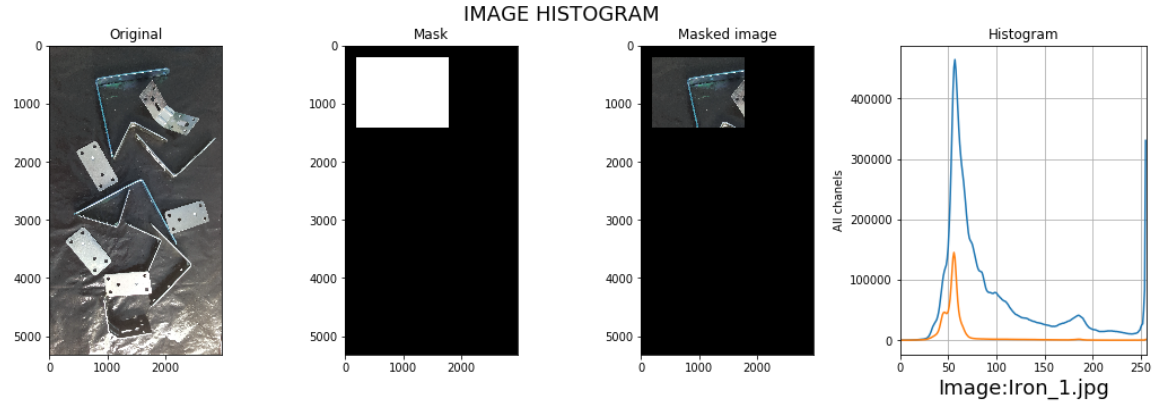

```
Cv2 Histogram for File: Copper_2.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Copper_
2.jpg
```



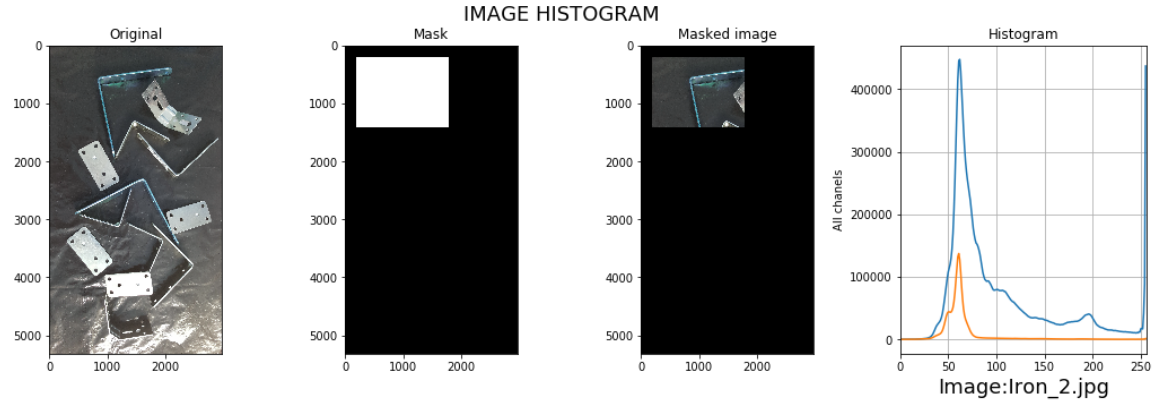

```
Cv2 Histogram for File: Copper_3.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Copper_
3.jpg
```



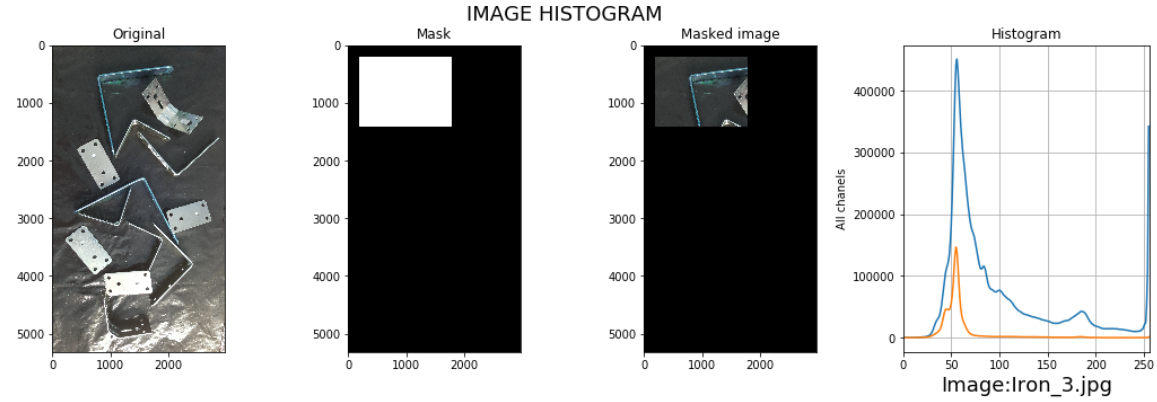

```
Cv2 Histogram for File: Iron_1.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Iron_1.
jpg
```



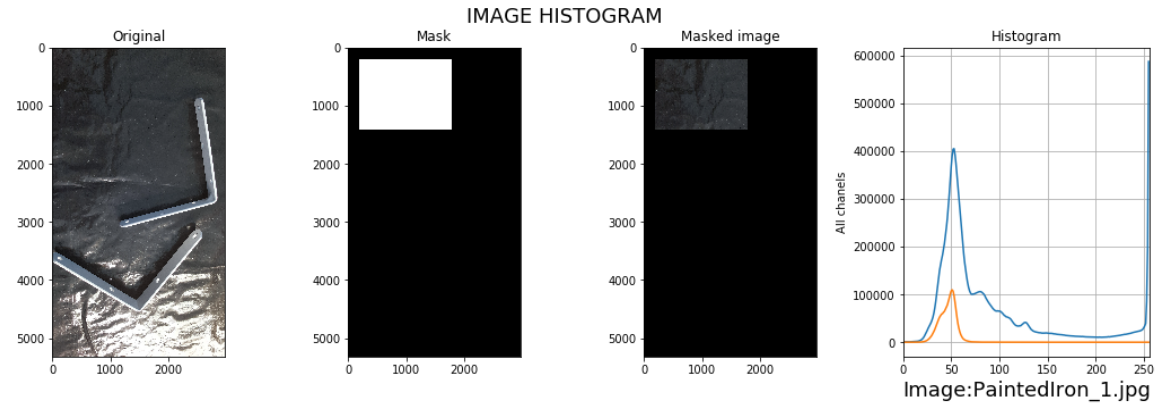

```
Cv2 Histogram for File: Iron_2.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Iron_2.
jpg
```



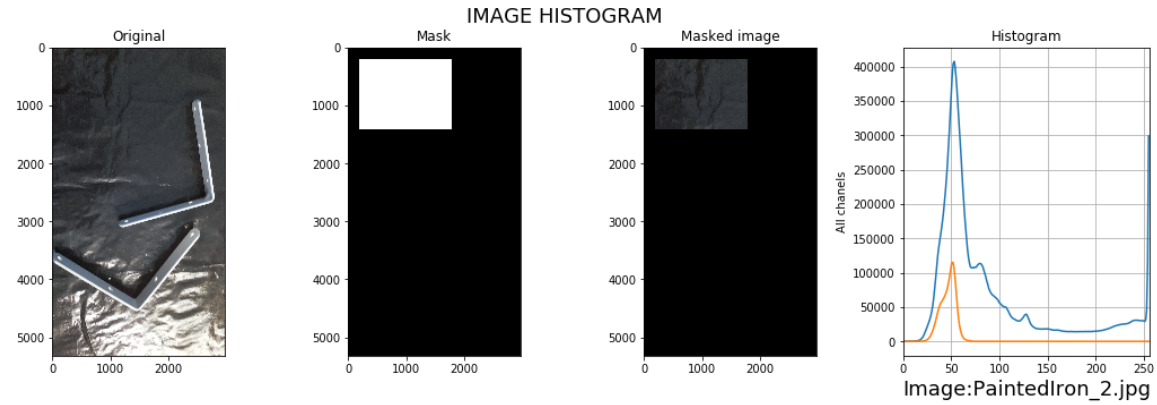

```
Cv2 Histogram for File: Iron_3.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Iron_3.
jpg
```



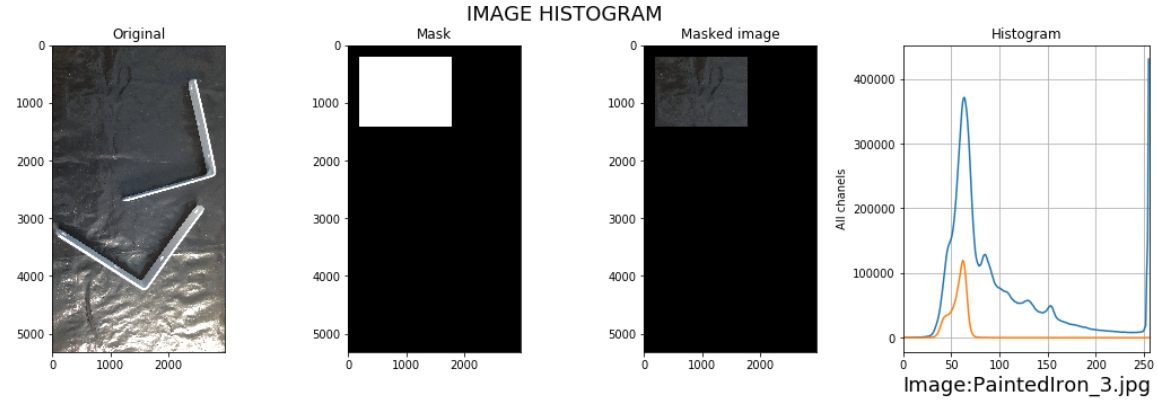

```
Cv2 Histogram for File: PaintedIron_1.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Painted
Iron_1.jpg
```



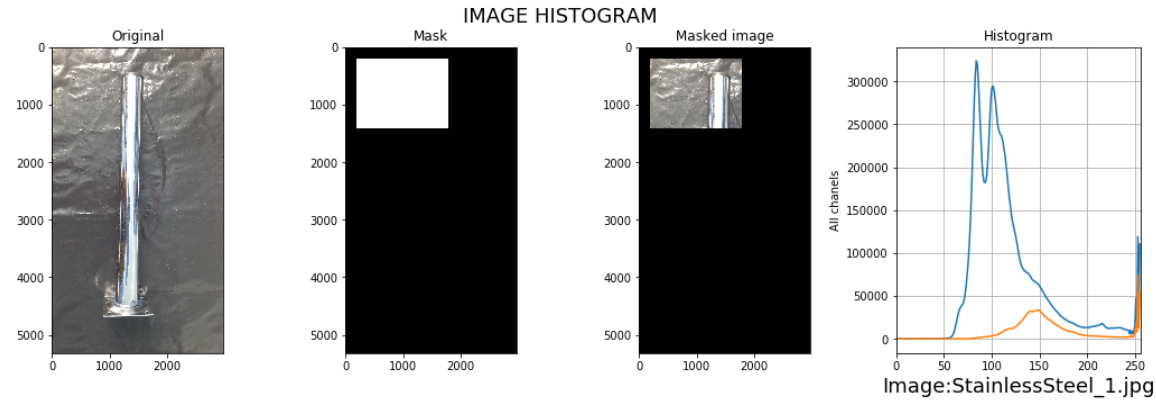

```
Cv2 Histogram for File: PaintedIron_2.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Painted
Iron_2.jpg
```



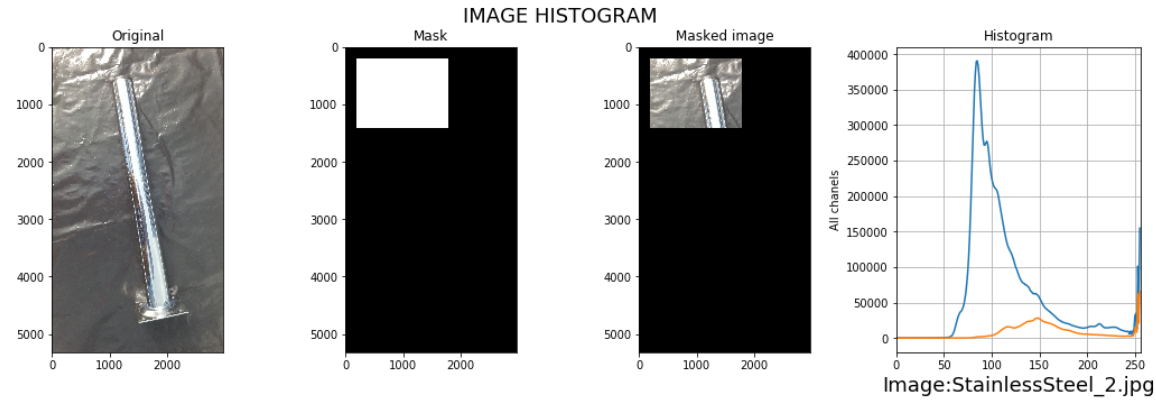

```
Cv2 Histogram for File: PaintedIron_3.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Painted
Iron_3.jpg
```




```
Cv2 Histogram for File: StainlessSteel_1.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Stainle
ssSteel_1.jpg
```




```
Cv2 Histogram for File: StainlessSteel_2.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Stainle
ssSteel_2.jpg
```




```
Cv2 Histogram for File: StainlessSteel_3.jpg
Cv2 Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portucalens
e\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Stainle
ssSteel_3.jpg
```

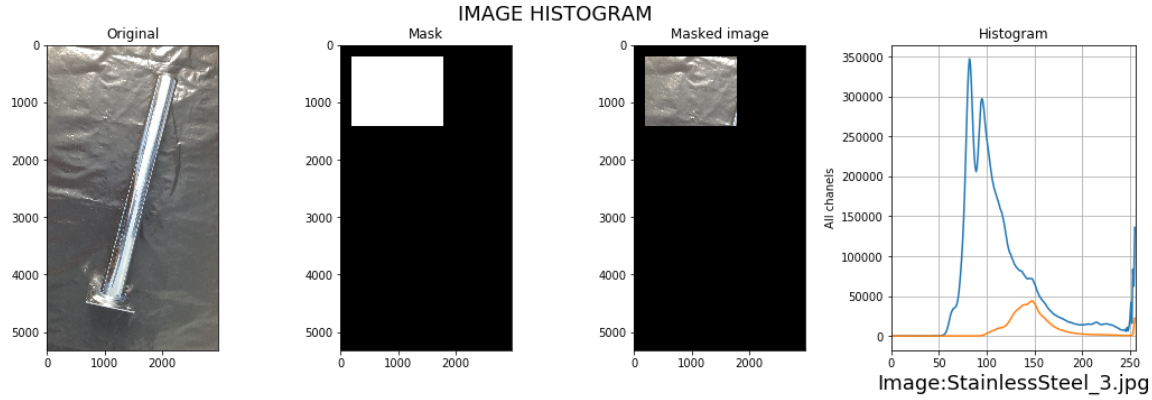
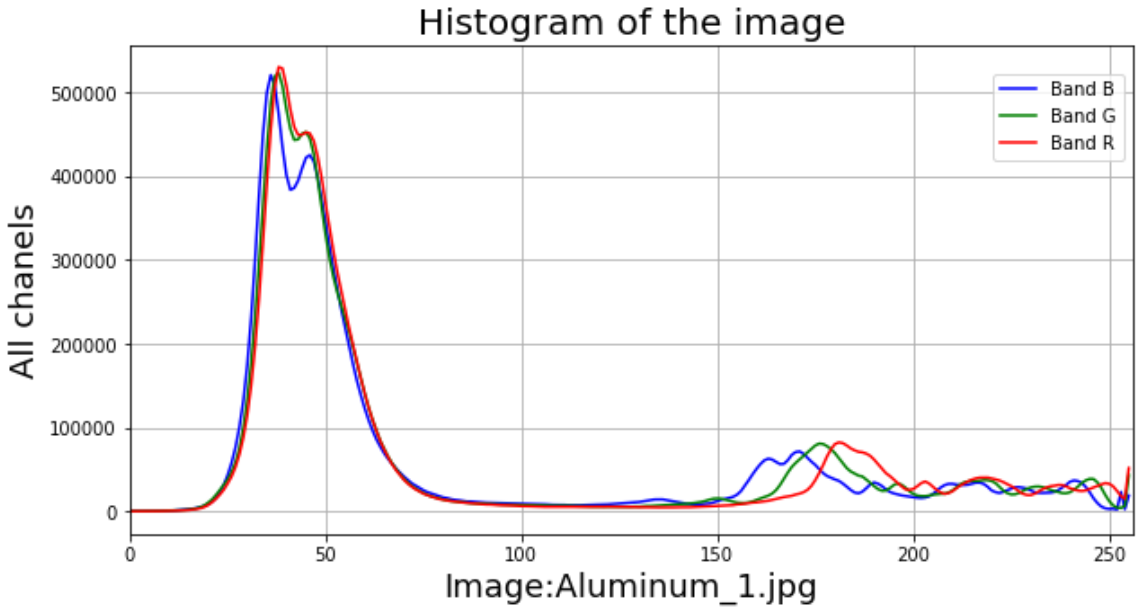


IMAGE HISTOGRAM
Original
Mask
Masked image
Histogram
All chanels
Image:StainlessSteel_3.jpg

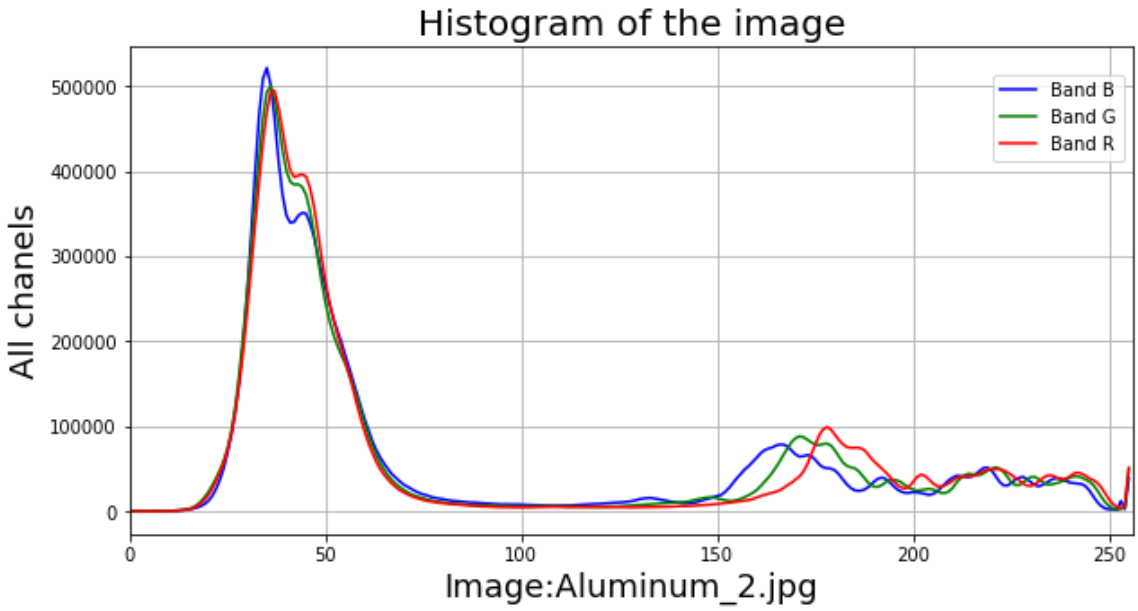
In [67]:

```
# HISTOGRAMS
# Print Histograms for all folder images
# list_of_images has all the name files

for x in list_of_images:
    print('Matplot Histogram for File:', x)
    print_matplot_hist(path, x)
```

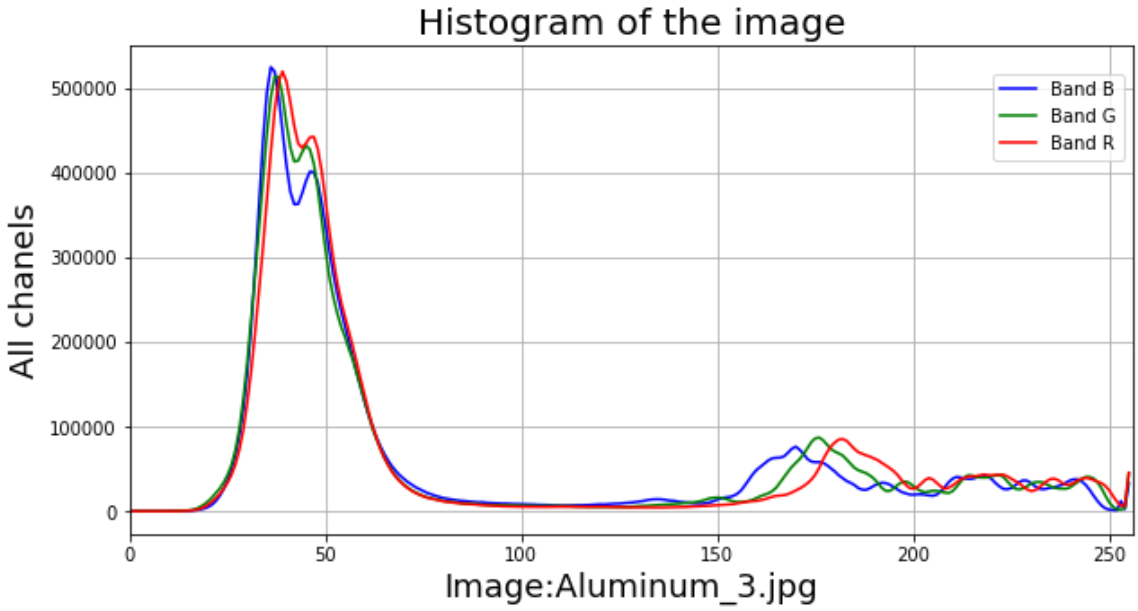
```
Matplot Histogram for File: Aluminum_1.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Alu
minum_1.jpg
```



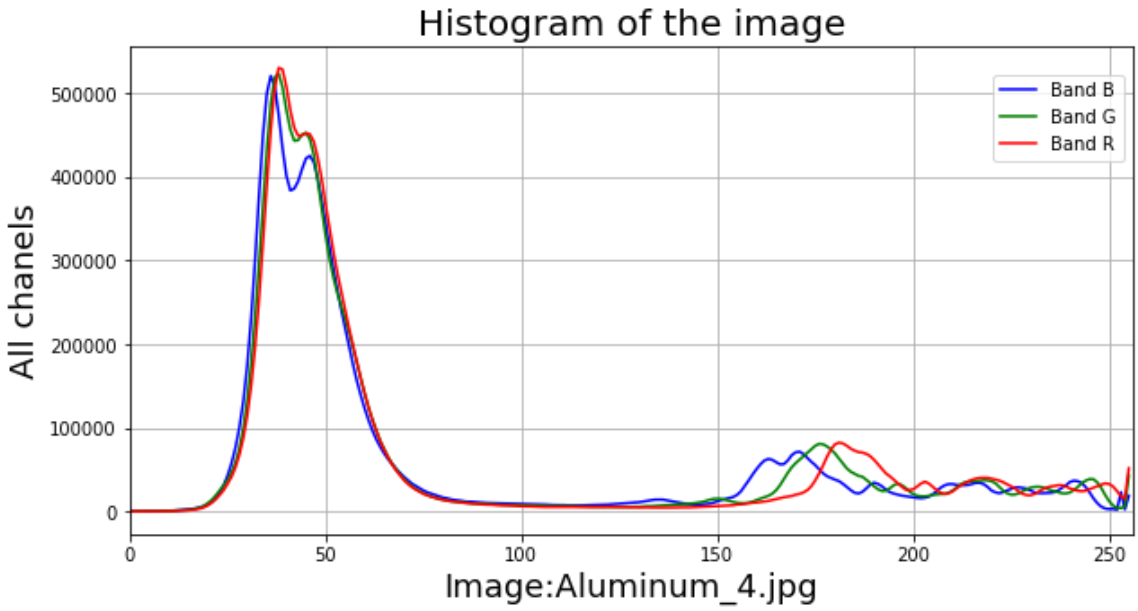

```
Matplot Histogram for File: Aluminum_2.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Alu
minum_2.jpg
```



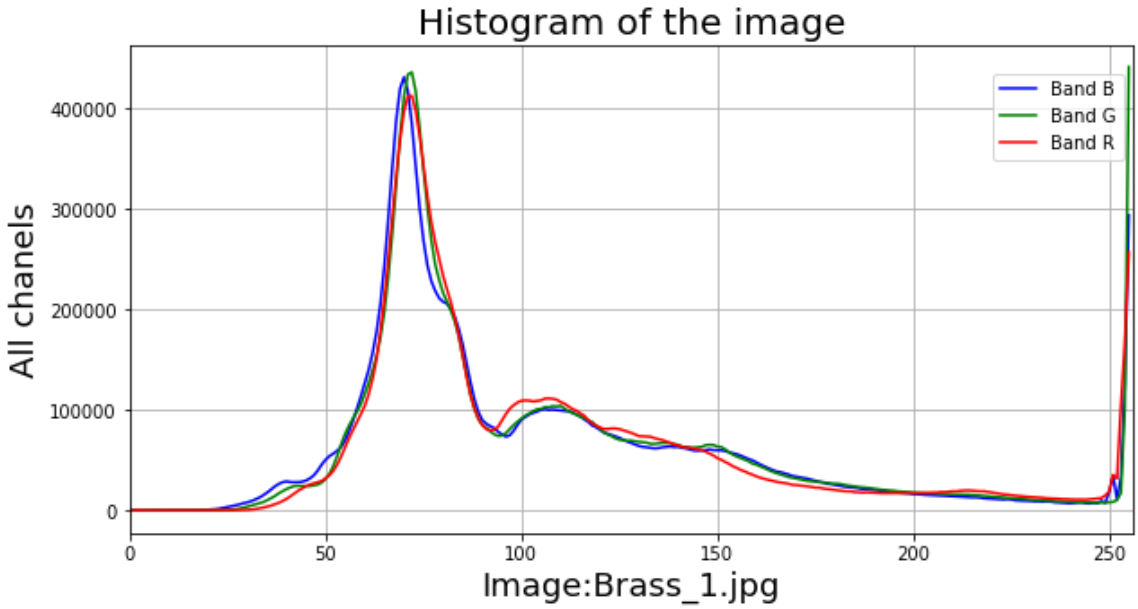
```
Matplot Histogram for File: Aluminum_3.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Alu
minum_3.jpg
```



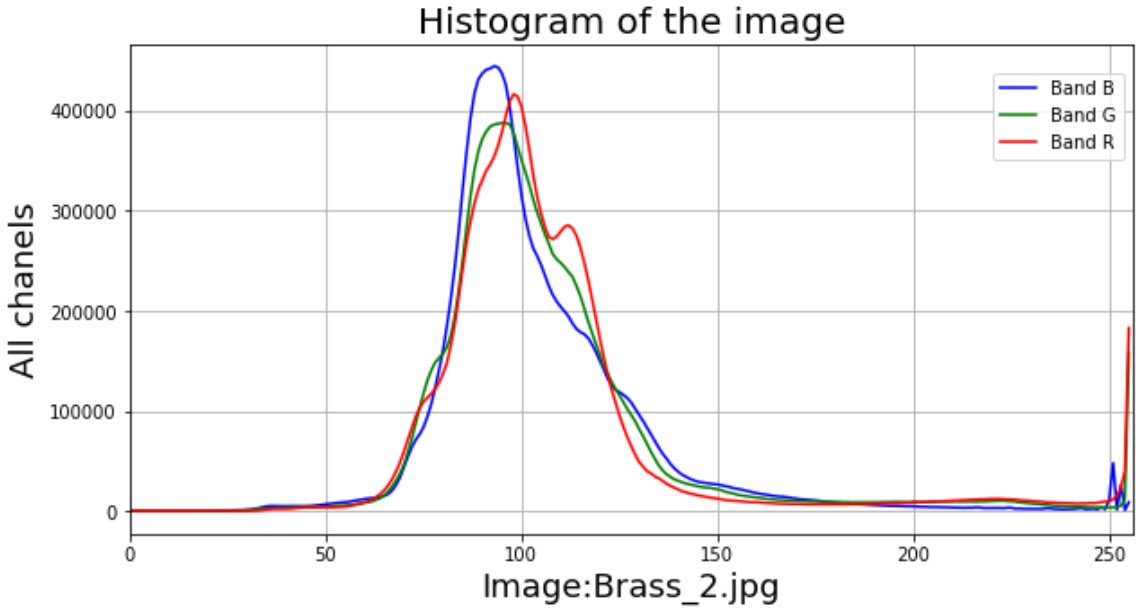

```
Matplot Histogram for File: Aluminum_4.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Alu
minum_4.jpg
```



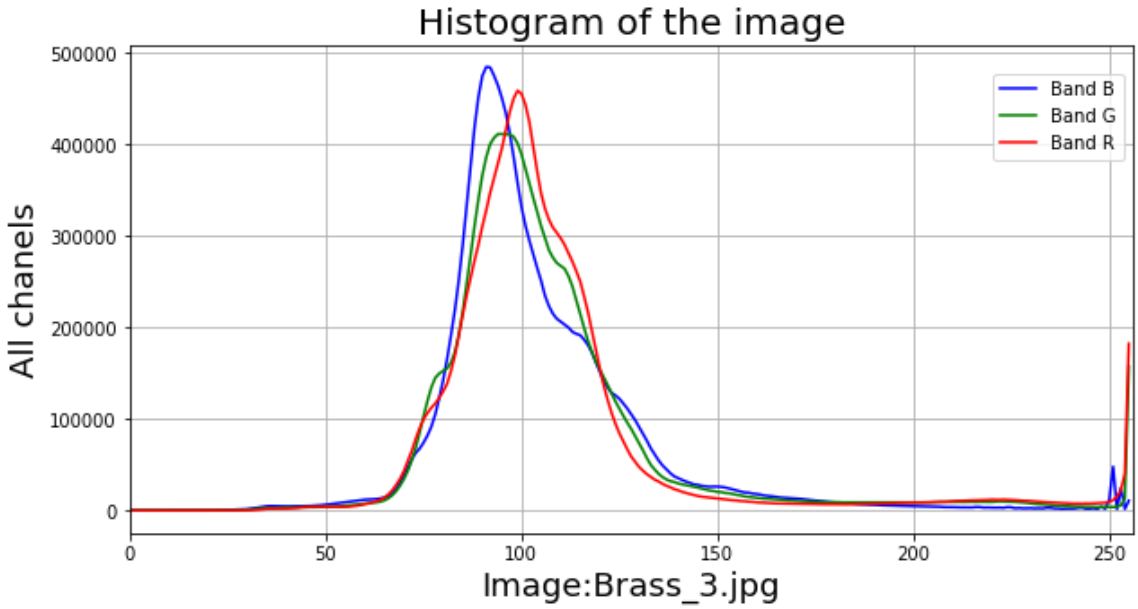

```
Matplot Histogram for File: Brass_1.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Bra
ss_1.jpg
```



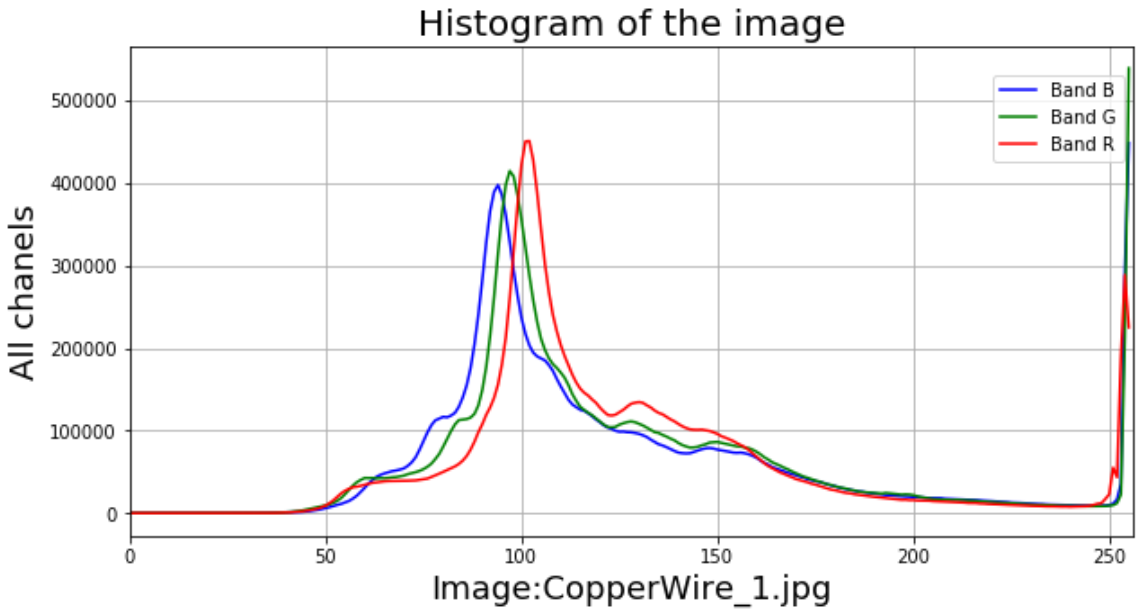

```
Matplot Histogram for File: Brass_2.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Bra
ss_2.jpg
```



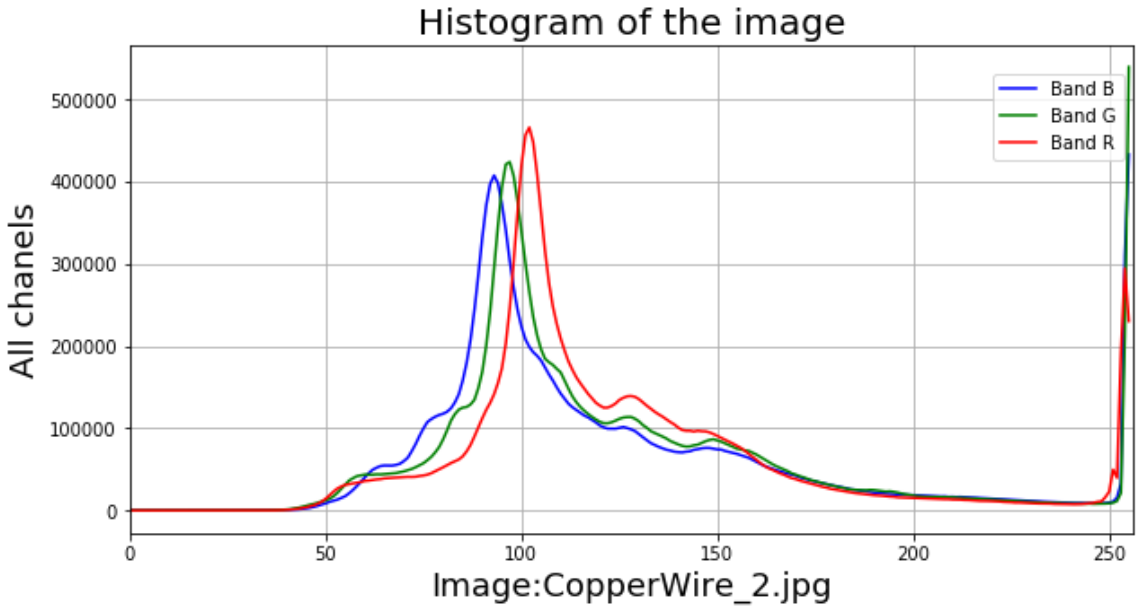

```
Matplot Histogram for File: Brass_3.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Bra
ss_3.jpg
```



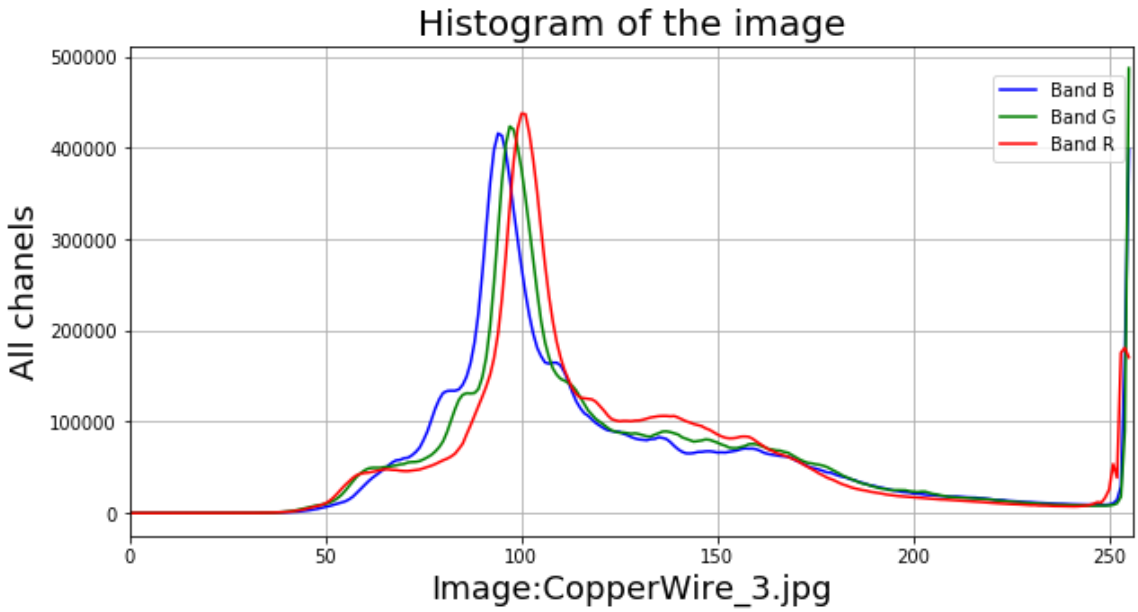

```
Matplot Histogram for File: CopperWire_1.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Cop
perWire_1.jpg
```



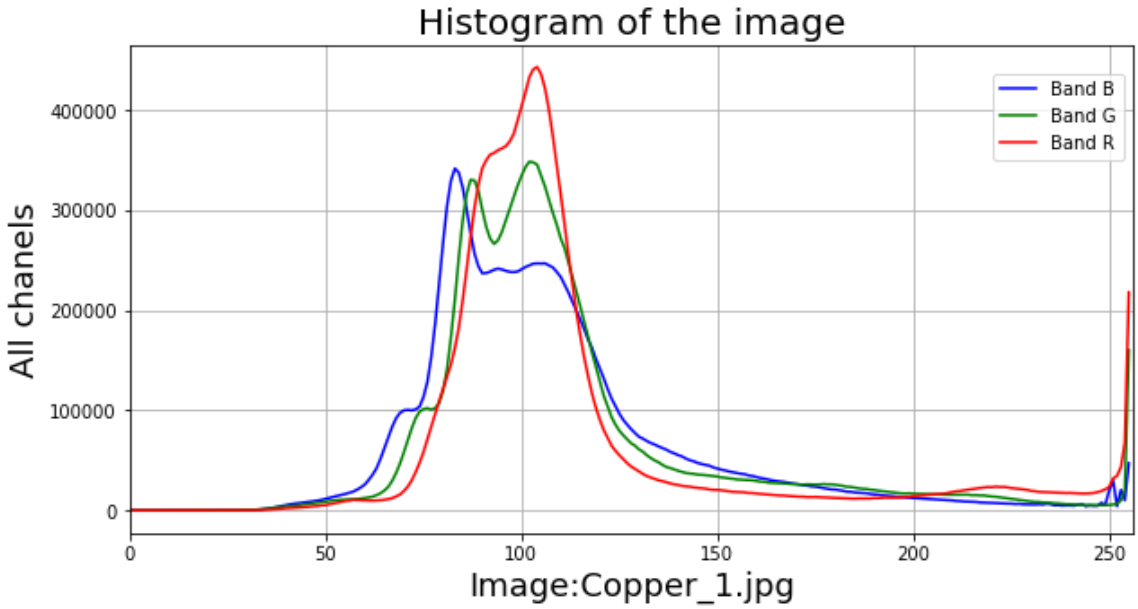

```
Matplot Histogram for File: CopperWire_2.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Cop
perWire_2.jpg
```



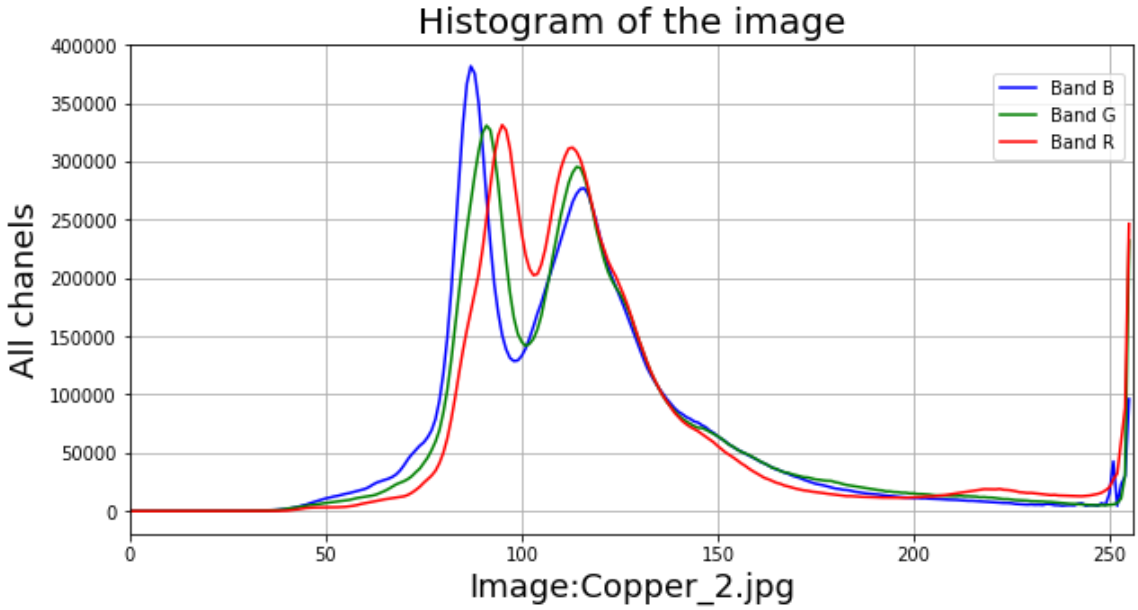

```
Matplot Histogram for File: CopperWire_3.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Cop
perWire_3.jpg
```



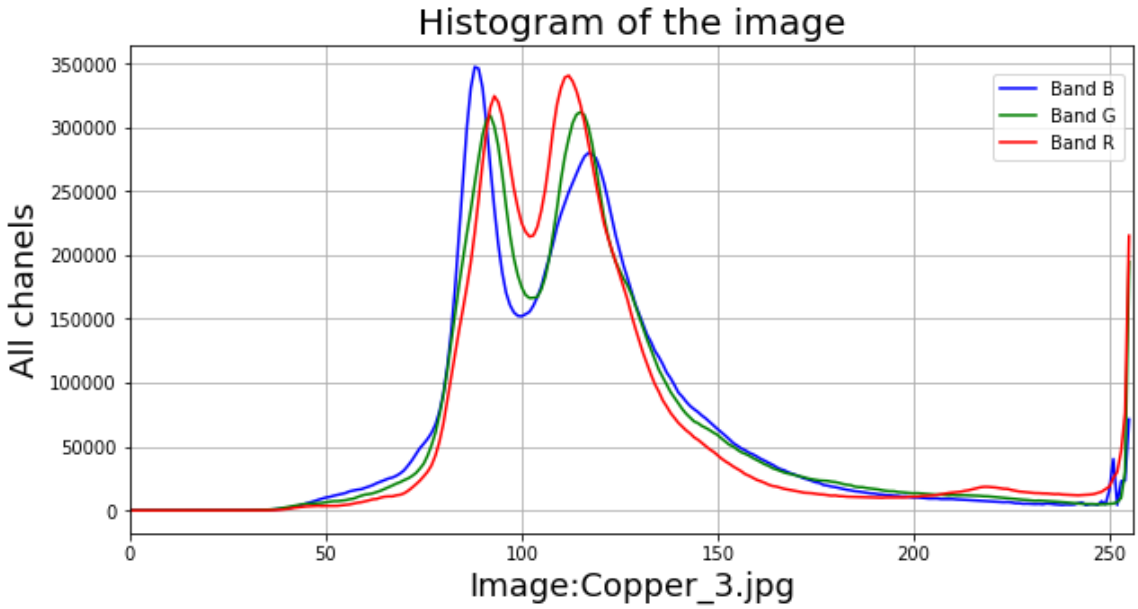

```
Matplot Histogram for File: Copper_1.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Cop
per_1.jpg
```




```
Matplot Histogram for File: Copper_2.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Cop
per_2.jpg
```

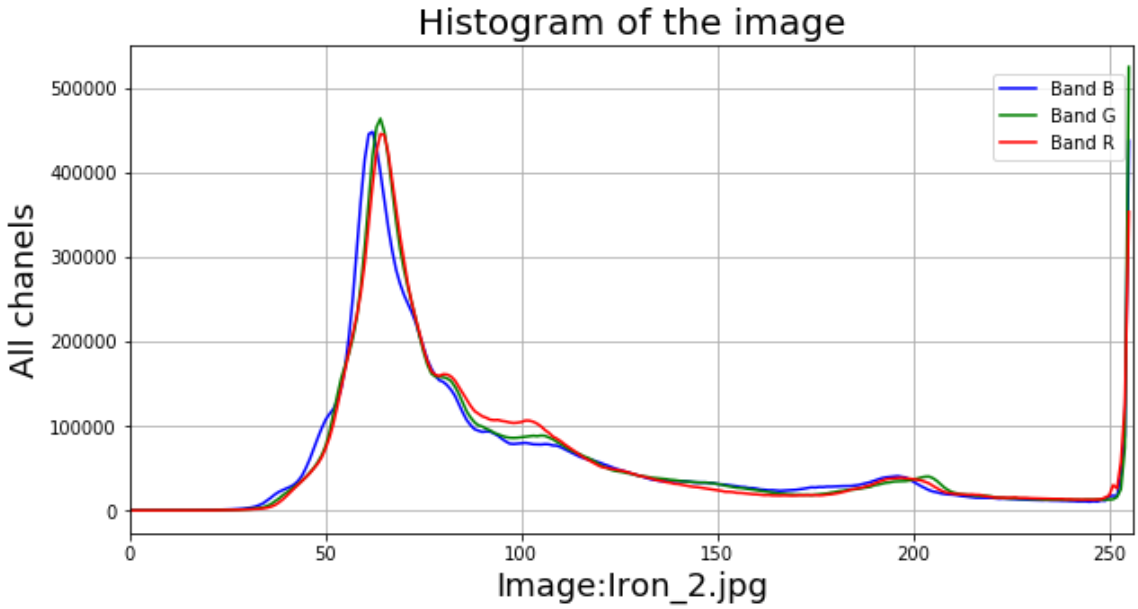



```
Matplot Histogram for File: Copper_3.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Cop
per_3.jpg
```

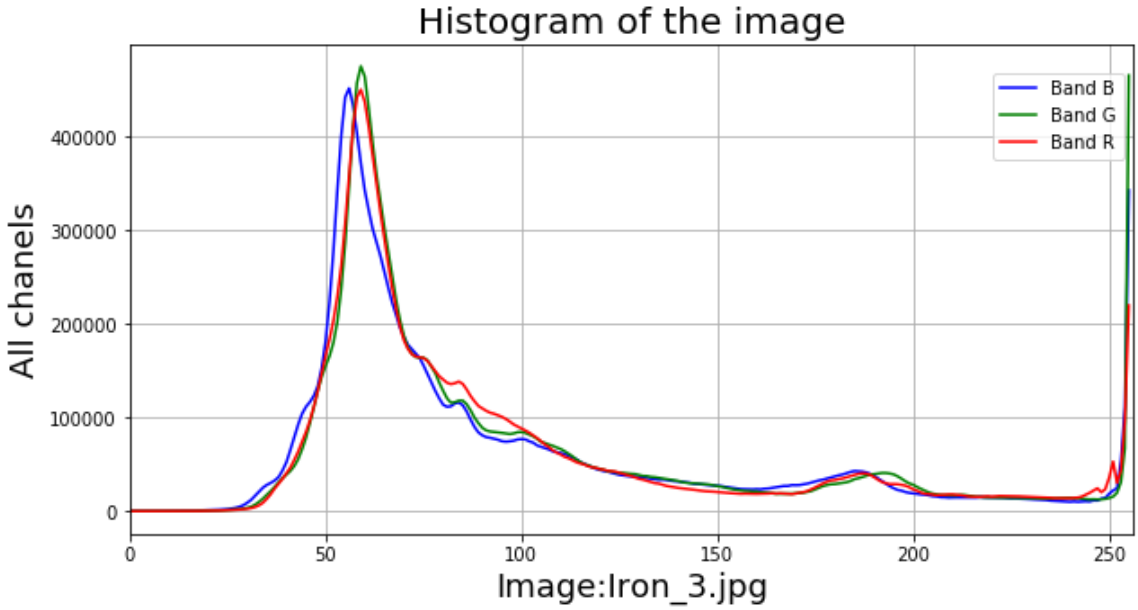



```
Matplot Histogram for File: Iron_1.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Iro
n_1.jpg
```

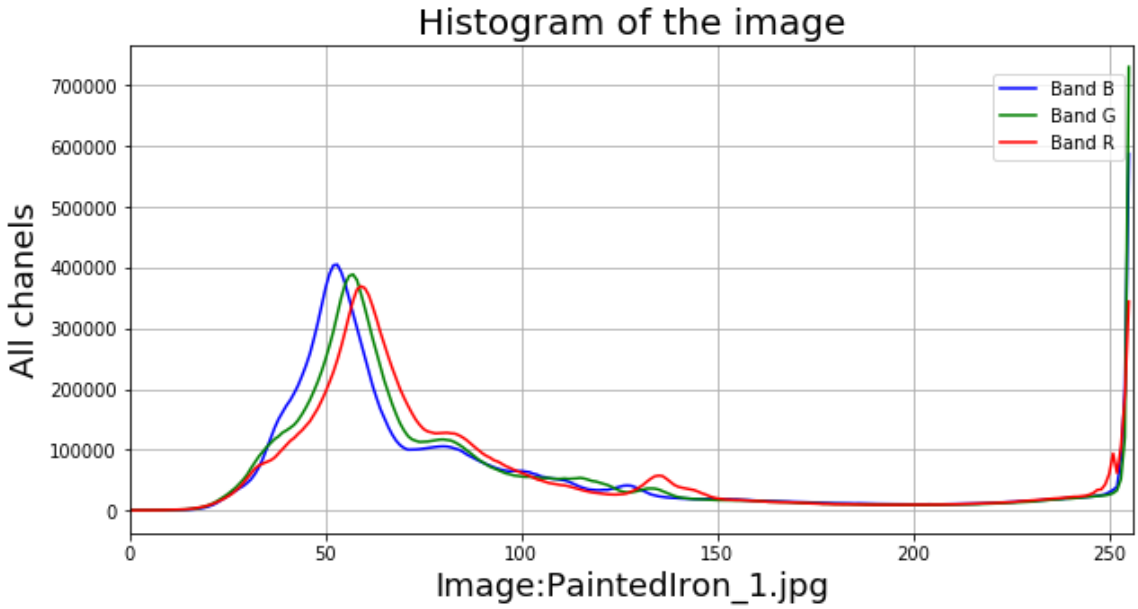

```
Matplot Histogram for File: Iron_2.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Iro
n_2.jpg
```




```
Matplot Histogram for File: Iron_3.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Iro
n_3.jpg
```

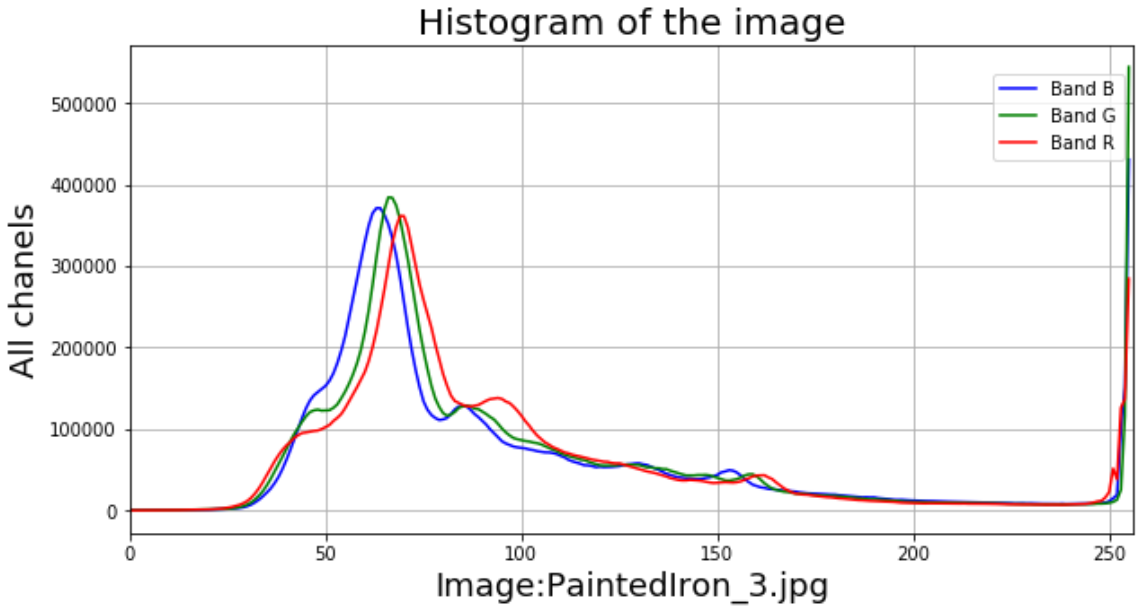



```
Matplot Histogram for File: PaintedIron_1.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Pai
ntedIron_1.jpg
```

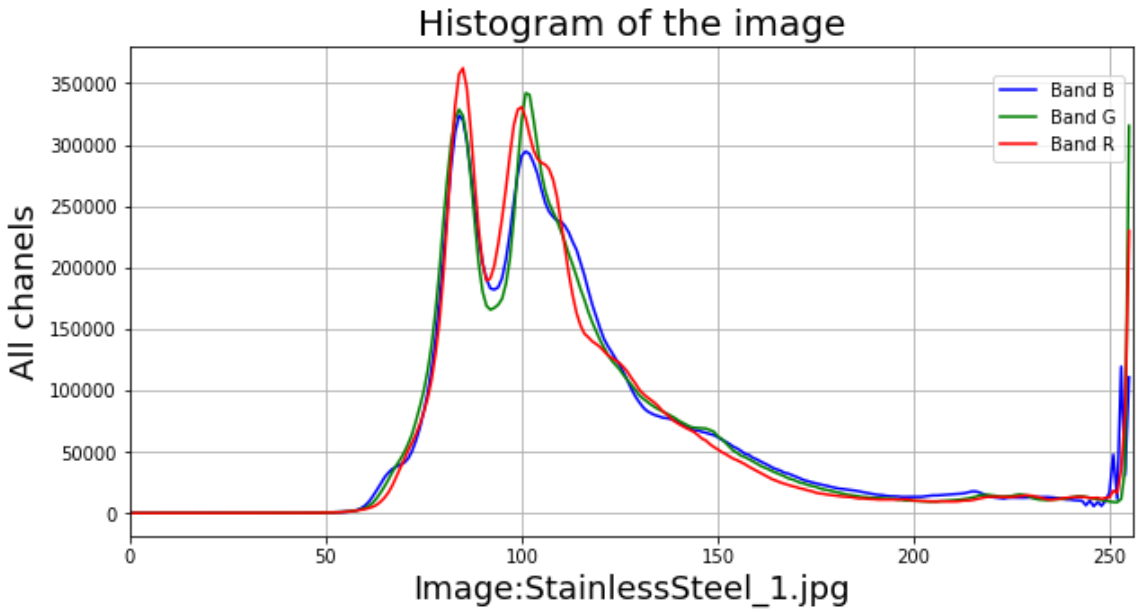



```
Matplot Histogram for File: PaintedIron_2.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Pai
ntedIron_2.jpg
```

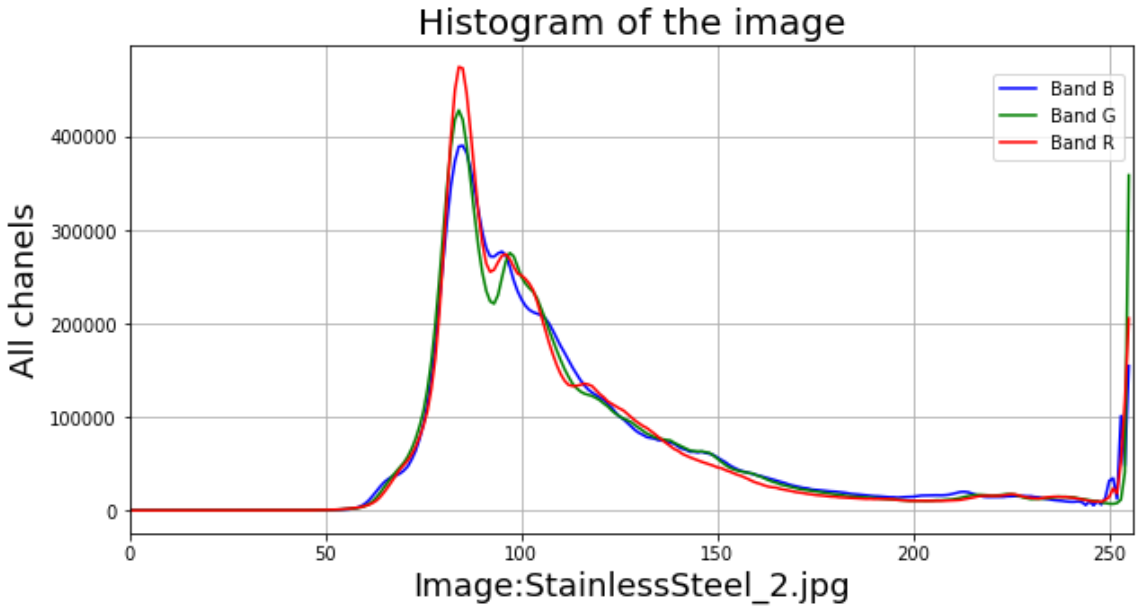

```
Matplot Histogram for File: PaintedIron_3.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Pai
ntedIron_3.jpg
```



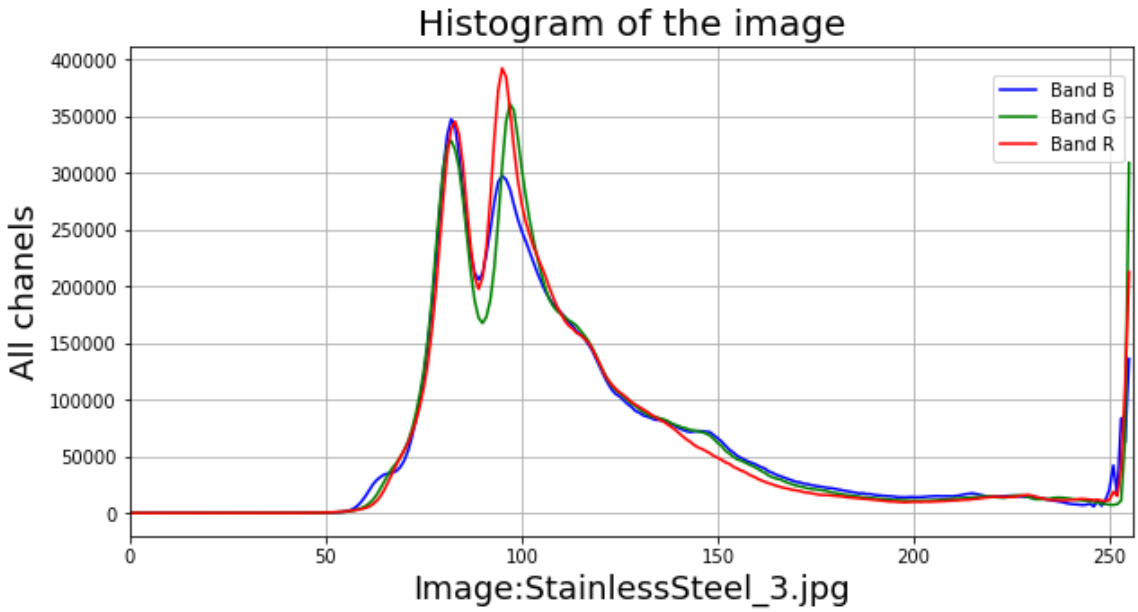

```
Matplot Histogram for File: StainlessSteel_1.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Sta
inlessSteel_1.jpg
```




```
Matplot Histogram for File: StainlessSteel_2.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Sta
inlessSteel_2.jpg
```




```
Matplot Histogram for File: StainlessSteel_3.jpg
Matplot Hist from file: C:\Users\manuel.robalinho\Google Drive\UPT_Portuca
lense\Trabalho final\Classificacao_Sucata\Jupyter_Notebook/imagedata07/Sta
inlessSteel_3.jpg
```



In [68]:

```
print('Finished')
```

```
Finished
```

In [ ]:

```
```