Data analisys from multi-spectral information

MRobalinho - UPT 27-4-2019

In [117]:

```
# Add libraries
import glob, os
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

In [118]:

```
# 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 [119]:

```
# Create path
xfile  = 'upt_data.xlsx'   # Excel file
xsheet = 'imagedata07'    # Excel Sheet

arq_xls = mypath + '/'+xfile
```

In [120]:

```
# Read Excel file
df = pd.read_excel(xfile,sheetname=xsheet,header=0,converters={'Size':str,'Color_RGB':s
tr})
```

In [121]:

```
df
```

Out[121]:

	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 [122]:

```
# Adjust Material name
df['Material'].replace('alum', 'aluminum',inplace=True)
df['Material'].replace('plasticised', 'plasticised_copper',inplace=True)
df['Material'].replace('stainless', 'stainless_steel',inplace=True)
df['Material'].replace('painted', 'painted_iron',inplace=True)
df
```

Out[122]:

	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 [123]:

```
df_plot = df.copy
```

In [124]:

```
# Create pivot table
df_plotx = df.groupby('File')['All_Bands', 'Sum_Ch0','Sum_Ch1','Sum_Ch2','Color_dec',
                                         'Med_Extrems', 'Max_Histog', 'Idx_Max_Histog','M
in_Histog',
                                         'Idx_Min_Histog','perc_R','perc_G','perc_B'].mea
n()
df_plotx
```

Out[124]:

File	All_Bands	Sum_Ch0	Sum_Ch1	Sum_Ch2	Color_dec	Med_Ex
Aluminum_1.jpg	3795.430722	1290.867334	1264.488693	1240.074695	16056319	127.5
Aluminum_2.jpg	4078.341415	1382.075655	1355.066322	1341.199438	16056319	127.5
Aluminum_3.jpg	3956.096638	1345.202894	1312.189751	1298.703993	16056319	127.5
Aluminum_4.jpg	3795.430722	1290.867334	1264.488693	1240.074695	16056319	127.5
Brass_1.jpg	809.915985	1720.902454	1712.269126	1671.711701	16646015	130.1
Brass_2.jpg	795.419913	1723.301362	1707.566316	1659.519531	16646015	134.1
Brass_3.jpg	808.952895	1728.201794	1713.305607	1662.412790	16646015	134.8
CopperWire_1.jpg	1701.825834	2024.420058	2000.821427	1971.551645	16646015	132.0
CopperWire_2.jpg	1631.193961	2008.148997	1975.862114	1942.150146	16646015	133.8
CopperWire_3.jpg	1636.447924	1991.258699	1982.614786	1957.541735	16646015	131.0
Copper_1.jpg	1037.993365	1828.316657	1783.668242	1720.975762	16646015	132.3
Copper_2.jpg	1393.597871	1951.851412	1897.481139	1839.232616	16646079	129.6
Copper_3.jpg	1339.507502	1905.950632	1882.785425	1845.738741	16646079	131.6
Iron_1.jpg	290.375493	1527.384117	1534.573907	1523.384765	16646015	127.5
Iron_2.jpg	563.473852	1618.699193	1631.213833	1608.528122	16646079	127.6
Iron_3.jpg	328.077431	1535.456848	1562.190463	1525.397416	16646015	127.8
PaintedIron_1.jpg	71.007131	1476.931962	1451.601158	1437.441307	16056319	127.5
PaintedIron_2.jpg	4287.385442	1441.211650	1429.451259	1416.722533	16056319	127.5
PaintedIron_3.jpg	335.612517	1546.023623	1548.193045	1536.363145	16056319	127.5
StainlessSteel_1.jpg	1256.493190	1838.259548	1852.298943	1860.901995	16056319	136.0
StainlessSteel_2.jpg	1159.233761	1805.040568	1821.808232	1827.352257	16056319	134.8
StainlessSteel_3.jpg	1193.991623	1814.545387	1837.406699	1837.006833	16056319	127.5

In [125]:

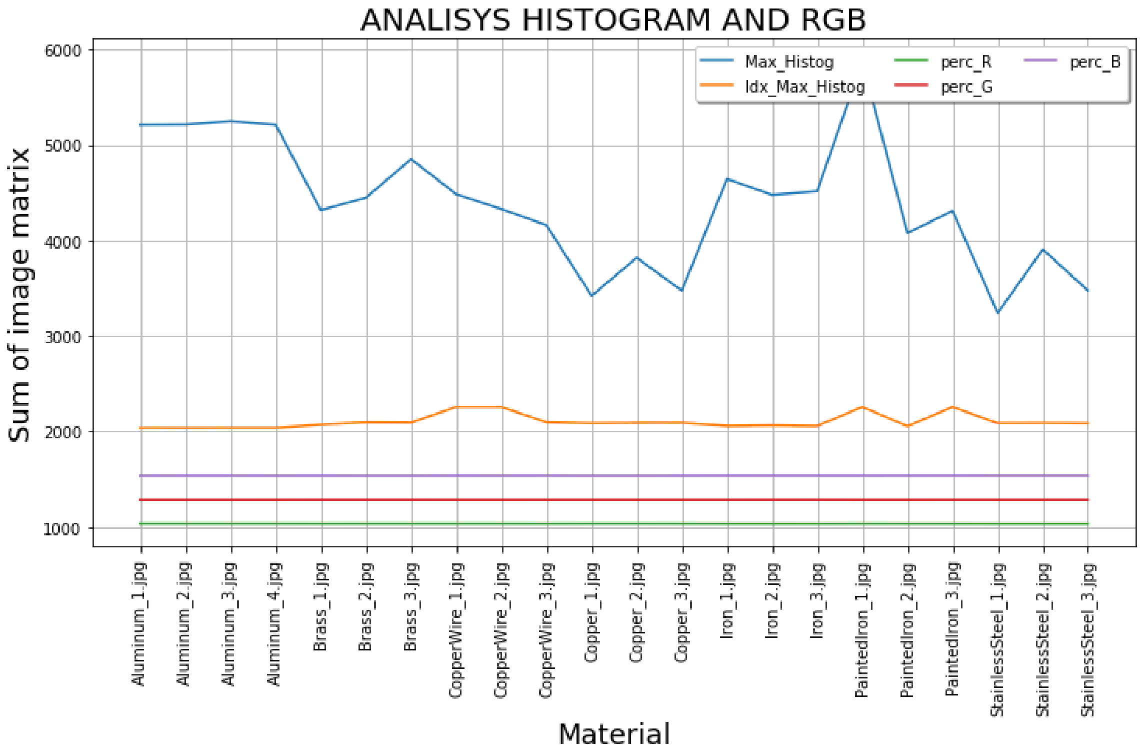
```
# Adjust values to better plot
df_plotx.Sum_Ch0        = df_plotx.Sum_Ch0 + 100 # to have diference lines during plot
df_plotx.Sum_Ch1        = df_plotx.Sum_Ch1 + 500
df_plotx.Sum_Ch2        = df_plotx.Sum_Ch2 + 1000
df_plotx.Color_dec      = df_plotx.Color_dec / 10000
#df_plotx.Color_dec      = df_plotx.Color_dec - 5000
df_plotx.Med_Extrems    = df_plotx.Med_Extrems * 10
df_plotx.Max_Histog     = df_plotx.Max_Histog
df_plotx.Idx_Max_Histog = df_plotx.Idx_Max_Histog + 1000
df_plotx.Min_Histog     = df_plotx.Min_Histog
df_plotx.Idx_Min_Histog = df_plotx.Idx_Min_Histog * 10

df_plotx.perc_R = df_plotx.perc_R + 1000
df_plotx.perc_G = df_plotx.perc_G + 1250
df_plotx.perc_B = df_plotx.perc_B + 1500

#df_plotx
```

In [126]:

```
# Adjust values to better plot
df_plotx.Max_Histog     = df_plotx.Max_Histog / 100
df_plotx.Idx_Max_Histog = df_plotx.Idx_Max_Histog + 1000
#df_plotx
```

In [127]:

```
df_plotx.plot(y=["Max_Histog","Idx_Max_Histog","perc_R", "perc_G","perc_B"],
figsize=(12,6), grid=True )
# Obtain legend (xticks) for X axis
loc_Array_sum = np.arange(len(df_plotx.index))
# Position of X labels
xtick_loc = list(loc_Array_sum)
# Name of x labels
xticks = list(df_plotx.index)
#-------
#plt.plot(df_plot.Array_sum)
plt.title('ANALISYS HISTOGRAM AND RGB',fontsize=20)
plt.ylabel('Sum of image matrix',fontsize=18)
plt.xticks(xtick_loc, df_plotx.index, rotation=90)
plt.xlabel('Material',fontsize=18)
plt.legend(loc='upper right', ncol=3, fancybox=True, shadow=True)
plt.savefig(xsheet+"_All Material Histogram and RGB.png")
plt.show()
```



In [128]:

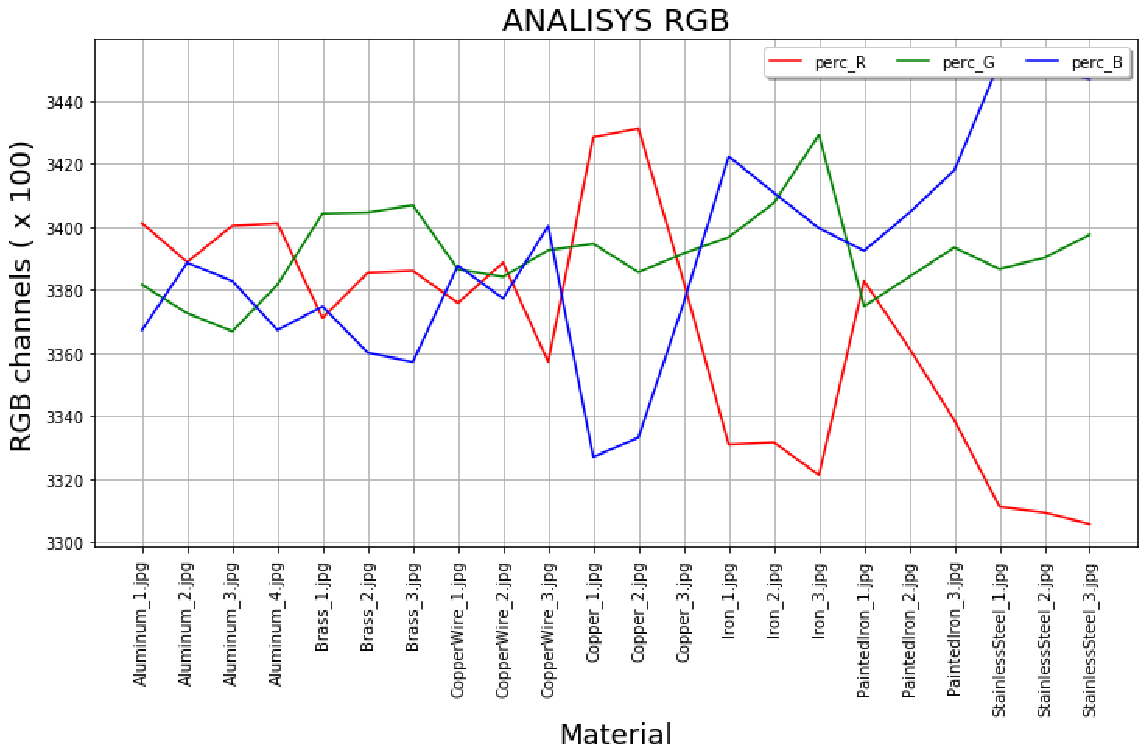
```
# Create pivot table
df_plotx = df.groupby('File')['All_Bands', 'Sum_Ch0','Sum_Ch1','Sum_Ch2','Color_dec',
                                        'Med_Extrems', 'Max_Histog', 'Idx_Max_Histog','M
in_Histog',
                                        'Idx_Min_Histog','perc_R','perc_G','perc_B'].mea
n()
#df_plotx
```

In [129]:

```
# Adjust values to better plot
df_plotx.perc_R     = df_plotx.perc_R * 100
df_plotx.perc_G     = df_plotx.perc_G * 100 + 50
df_plotx.perc_B     = df_plotx.perc_B * 100 + 100
```

In [130]:

```
df_plotx.plot(y=["perc_R", "perc_G","perc_B"],
figsize=(12,6), grid=True, color=('r','g','b') )
# Obtain legend (xticks) for X axis
loc_Array_sum = np.arange(len(df_plotx.index))
# Position of X labels
xtick_loc = list(loc_Array_sum)
# Name of x labels
xticks = list(df_plotx.index)
#-------
#plt.plot(df_plot.Array_sum)
plt.title('ANALISYS RGB',fontsize=20)
plt.ylabel('RGB channels ( x 100)',fontsize=18)
plt.xticks(xtick_loc, df_plotx.index, rotation=90)
plt.xlabel('Material',fontsize=18)
plt.legend(loc='upper right', ncol=3, fancybox=True, shadow=True)
plt.savefig(xsheet+"_All material Analisys RGB.png")
plt.show()
```



In [131]:

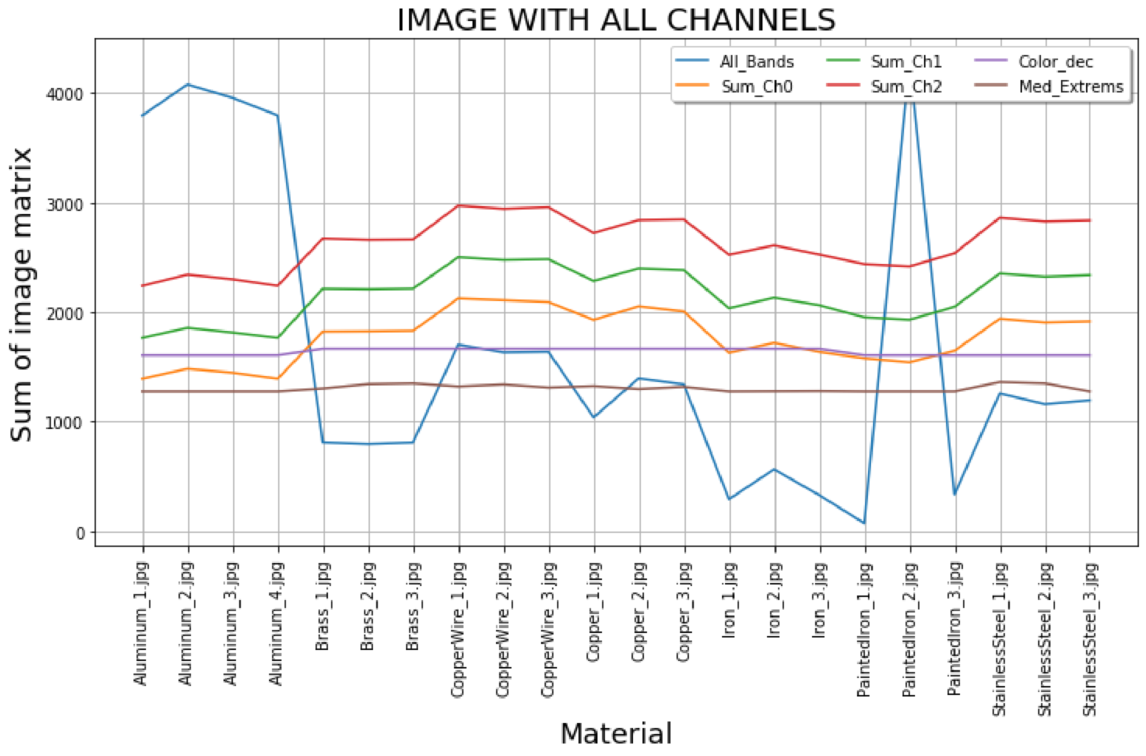
```
# Create pivot table
df_plotx = df.groupby('File')['All_Bands', 'Sum_Ch0','Sum_Ch1','Sum_Ch2','Color_dec',
                                        'Med_Extrems', 'Max_Histog', 'Idx_Max_Histog','M
in_Histog',
                                        'Idx_Min_Histog','perc_R','perc_G','perc_B'].mea
n()
#df_plotx
```

In [132]:

```
# Adjust values to better plot
df_plotx.Sum_Ch0        = df_plotx.Sum_Ch0 + 100 # to have diference lines during plot
df_plotx.Sum_Ch1        = df_plotx.Sum_Ch1 + 500
df_plotx.Sum_Ch2        = df_plotx.Sum_Ch2 + 1000
df_plotx.Color_dec      = df_plotx.Color_dec / 10000
#df_plotx.Color_dec      = df_plotx.Color_dec - 5000
df_plotx.Med_Extrems    = df_plotx.Med_Extrems * 10
df_plotx.Max_Histog     = df_plotx.Max_Histog
df_plotx.Idx_Max_Histog = df_plotx.Idx_Max_Histog + 1000
df_plotx.Min_Histog     = df_plotx.Min_Histog
df_plotx.Idx_Min_Histog = df_plotx.Idx_Min_Histog * 10

df_plotx.perc_R = df_plotx.perc_R + 1000
df_plotx.perc_G = df_plotx.perc_G + 1250
df_plotx.perc_B = df_plotx.perc_B + 1500

#df_plotx
```

In [133]:

```
df_plotx.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_plotx.index))
# Position of X labels
xtick_loc = list(loc_Array_sum)
# Name of x labels
xticks = list(df_plotx.index)
#-------
#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_plotx.index, rotation=90)
plt.xlabel('Material',fontsize=18)
plt.legend(loc='upper right', ncol=3, fancybox=True, shadow=True)
plt.savefig(xsheet+"_All material Line Graph all channels information.png")
plt.show()
```



In [134]:

```
'''
# Adjust values to better plot
df_plot.Sum_Ch0        = df_plot.Sum_Ch0 + 100 # to have diference lines during plot
df_plot.Sum_Ch1        = df_plot.Sum_Ch1 + 500
df_plot.Sum_Ch2        = df_plot.Sum_Ch2 + 1000
df_plot.Color_dec      = df_plot.Color_dec / 1000
df_plot.Color_dec      = df_plot.Color_dec - 5000
df_plot.Med_Extrems    = df_plot.Med_Extrems * 10
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[134]:

```
'\n# Adjust values to better plot\ndf_plot.Sum_Ch0        = df_plot.Sum_Ch
0 + 100 # to have diference lines during plot\ndf_plot.Sum_Ch1        = df
_plot.Sum_Ch1 + 500\ndf_plot.Sum_Ch2        = df_plot.Sum_Ch2 + 1000\ndf_p
lot.Color_dec      = df_plot.Color_dec / 1000\ndf_plot.Color_dec      = df
_plot.Color_dec - 5000\ndf_plot.Med_Extrems    = df_plot.Med_Extrems * 10
\ndf_plot.Max_Histog     = df_plot.Max_Histog / 1000\ndf_plot.Idx_Max_Hist
og = df_plot.Idx_Max_Histog + 1000\ndf_plot.Min_Histog     = df_plot.Min_H
istog * 100\ndf_plot.Idx_Min_Histog = df_plot.Idx_Min_Histog * 10\n\ndf_pl
ot.perc_R = df_plot.perc_R + 1000\ndf_plot.perc_G = df_plot.perc_G + 1100
\ndf_plot.perc_B = df_plot.perc_B + 1200\n\ndf_plot\n'
```

In [135]:

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

Out[135]:

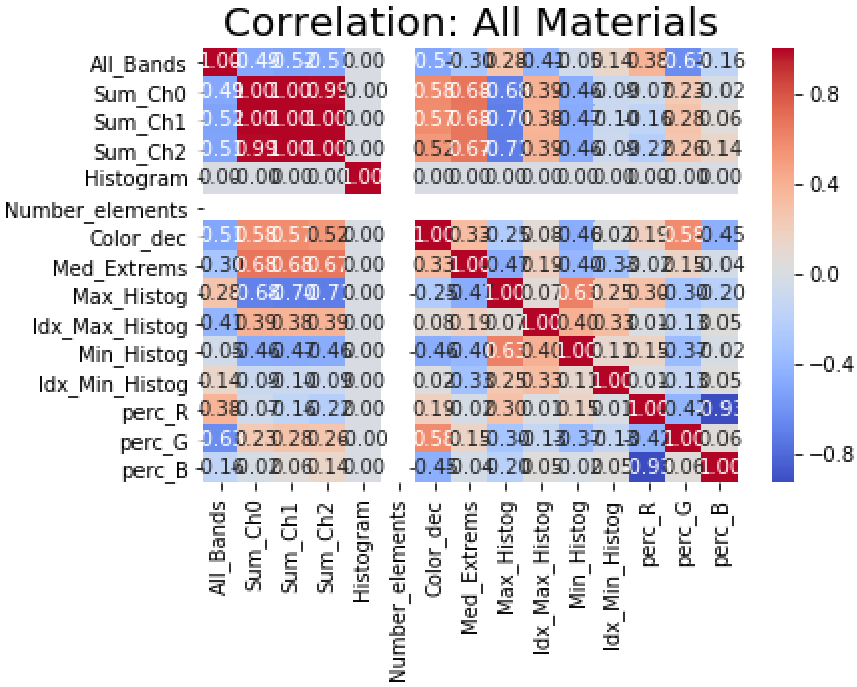
	All_Bands	Sum_Ch0	Sum_Ch1	Sum_Ch2	Color_dec	Med_Extre
Material						
Aluminum	3906.324874	1327.253304	1299.058365	1280.013205	1.605632e+07	127.500
Brass	804.762931	1724.135203	1711.047016	1664.548007	1.664602e+07	133.055
Copper	1257.032913	1895.372900	1854.644935	1801.982373	1.664606e+07	131.222
CopperWire	1656.489240	2007.942585	1986.432776	1957.081175	1.664602e+07	132.277
Iron	393.975592	1560.513386	1575.992734	1552.436768	1.664604e+07	127.666
PaintedIron	1564.668363	1488.055745	1476.415154	1463.508995	1.605632e+07	127.500
StainlessSteel	1203.239525	1819.281834	1837.171291	1841.753695	1.605632e+07	132.777

In [136]:

```
# Plot Correlation
def plot_corr(xdf,material,xsheet):
    x,y = 10,5

    fig, ax = plt.subplots()
    sns.heatmap(xdf.corr(method='pearson'), annot=True, fmt='.2f',
                cmap=plt.get_cmap('coolwarm'), cbar=True, ax=ax)

    ax.set_yticklabels(ax.get_yticklabels(), rotation="horizontal")
    plt.title('Correlation: '+material,fontsize=20)
    plt.savefig(xsheet+'_Corr_'+material+'.png', bbox_inches='tight', pad_inches=0.0)
```

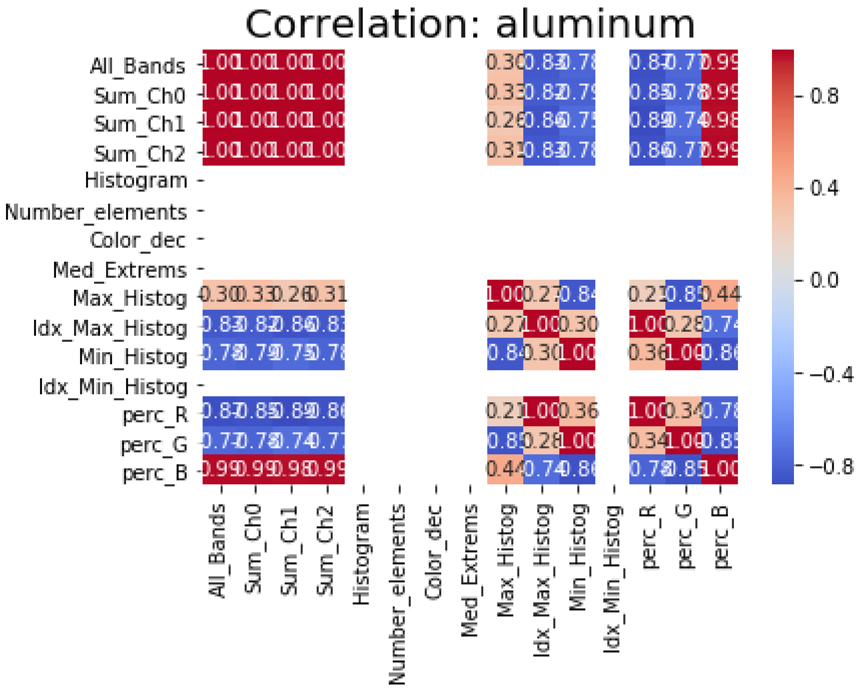
In [137]:

```
material = 'All'
df_x = df

# Plot Correlation
plot_corr(df_x,'All Materials',xsheet)
```



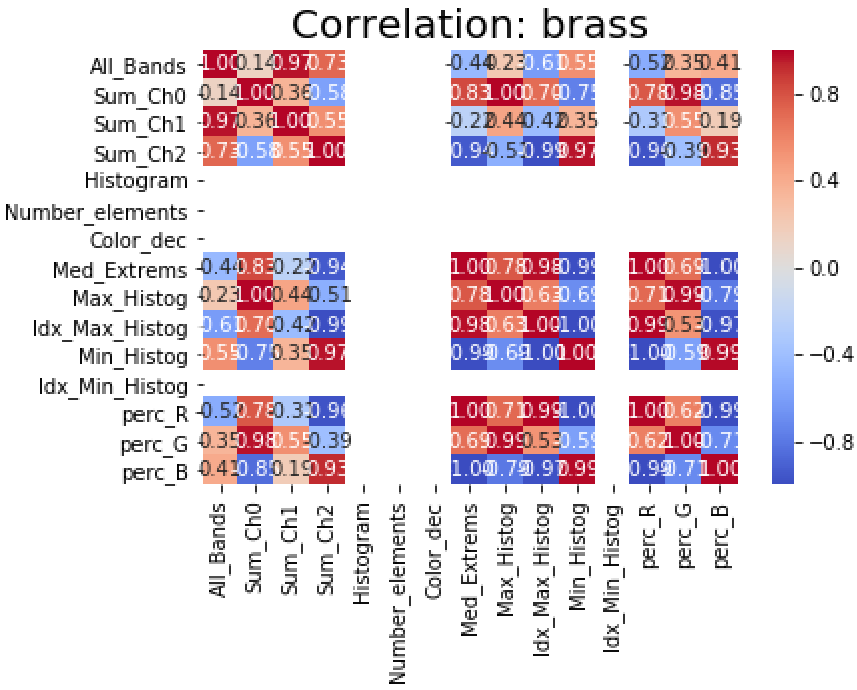
In [138]:

```
material = 'Aluminum'
df_x = df[df['Material']== material]

# Plot Correlation
plot_corr(df_x,'aluminum',xsheet)
```



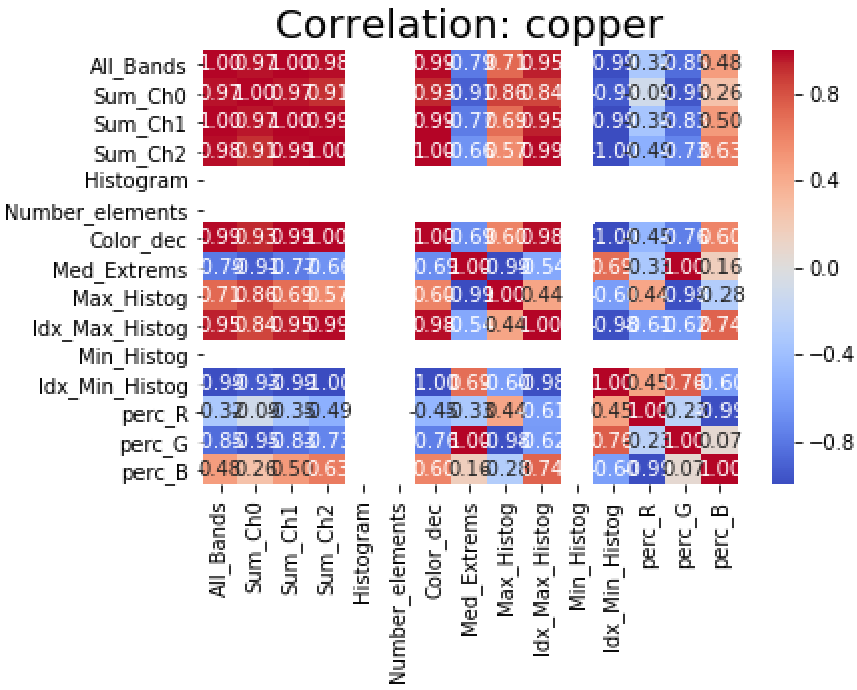
In [139]:

```
material = 'Brass'
df_x = df[df['Material']== material]

# Plot Correlation
plot_corr(df_x,'brass',xsheet)
```



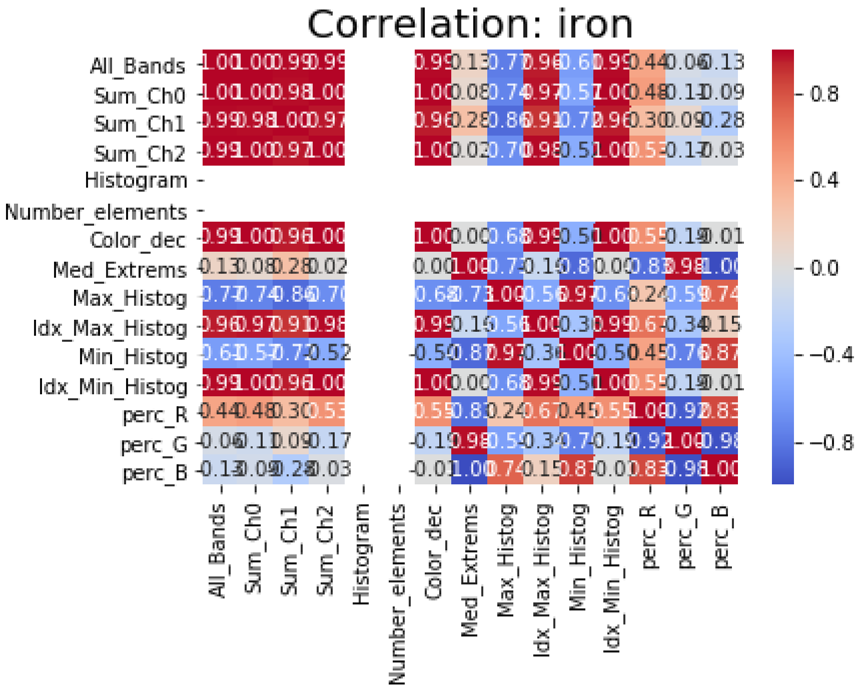
In [140]:

```
material = 'Copper'
df_x = df[df['Material']== material]

# Plot Correlation
plot_corr(df_x,'copper',xsheet)
```



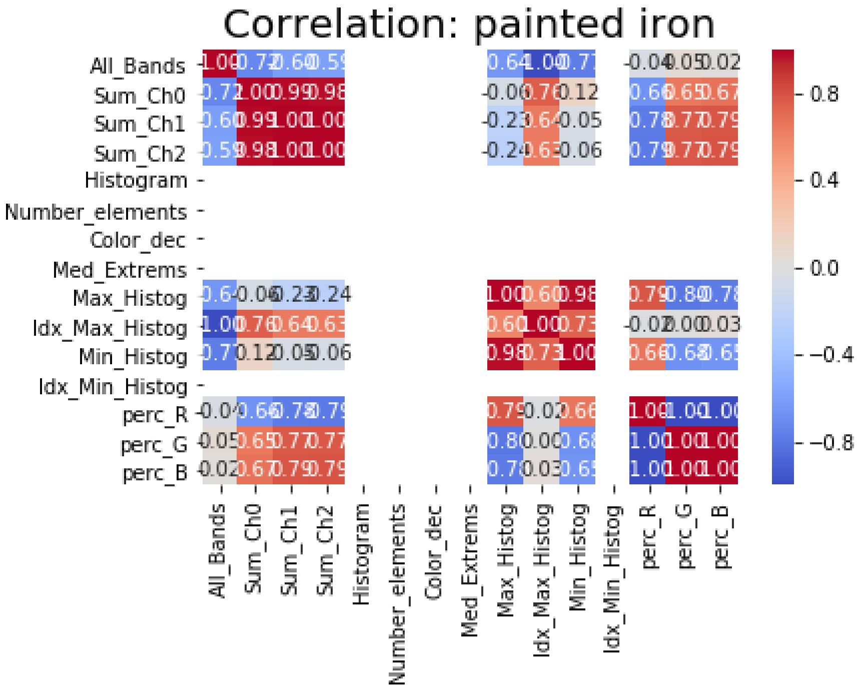
In [141]:

```
material = 'Iron'
df_x = df[df['Material']== material]

# Plot Correlation
plot_corr(df_x,'iron',xsheet)
```



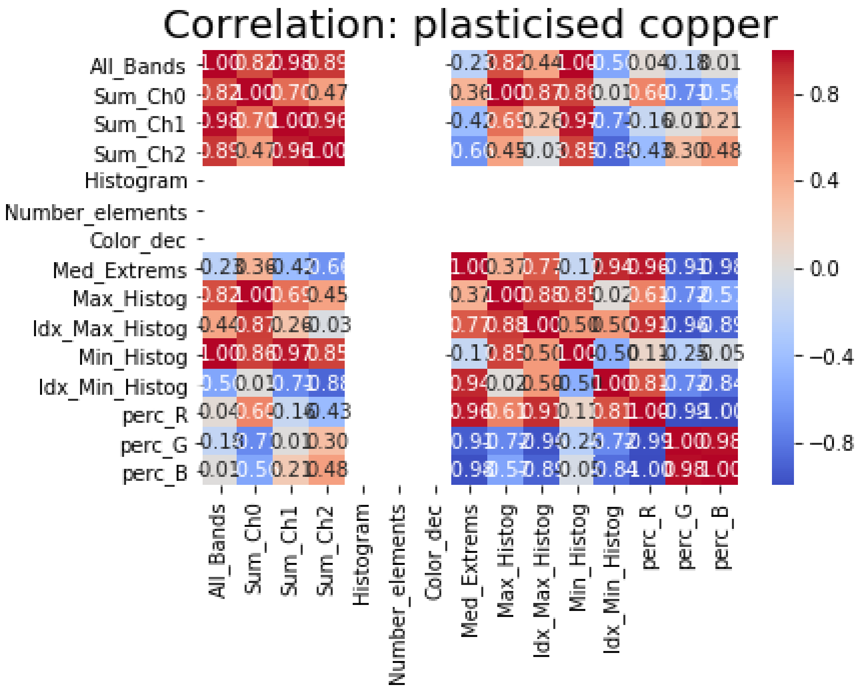
In [142]:

```
material = 'PaintedIron'
df_x = df[df['Material']== material]

# Plot Correlation
plot_corr(df_x,'painted iron',xsheet)
```



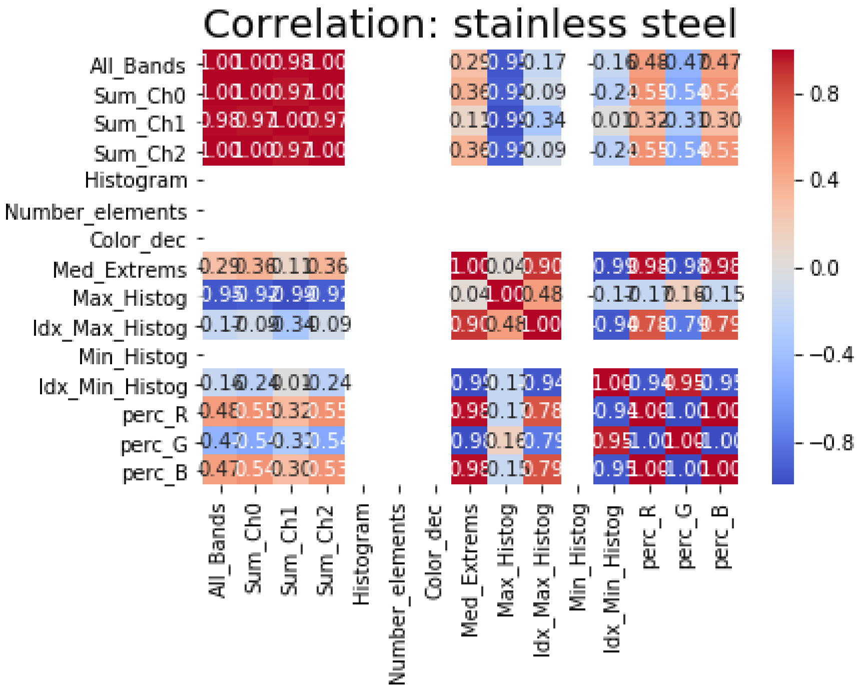
In [143]:

```
material = 'CopperWire'
df_x = df[df['Material']== material]
# Plot Correlation
plot_corr(df_x,'plasticised copper',xsheet)
```



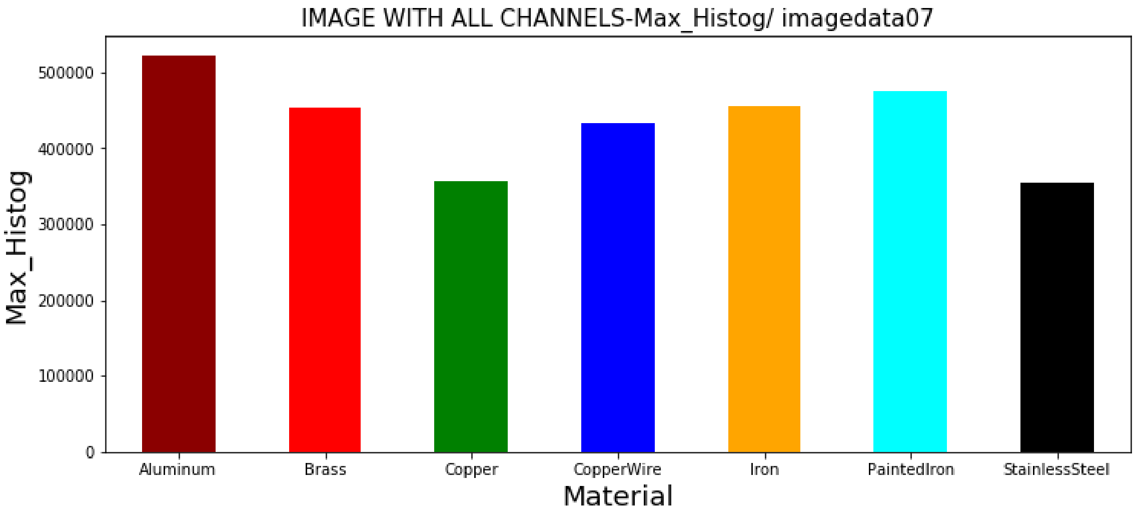
In [144]:

```
material = 'StainlessSteel'
df_x = df[df['Material']== material]

# Plot Correlation
plot_corr(df_x,'stainless steel',xsheet)
```



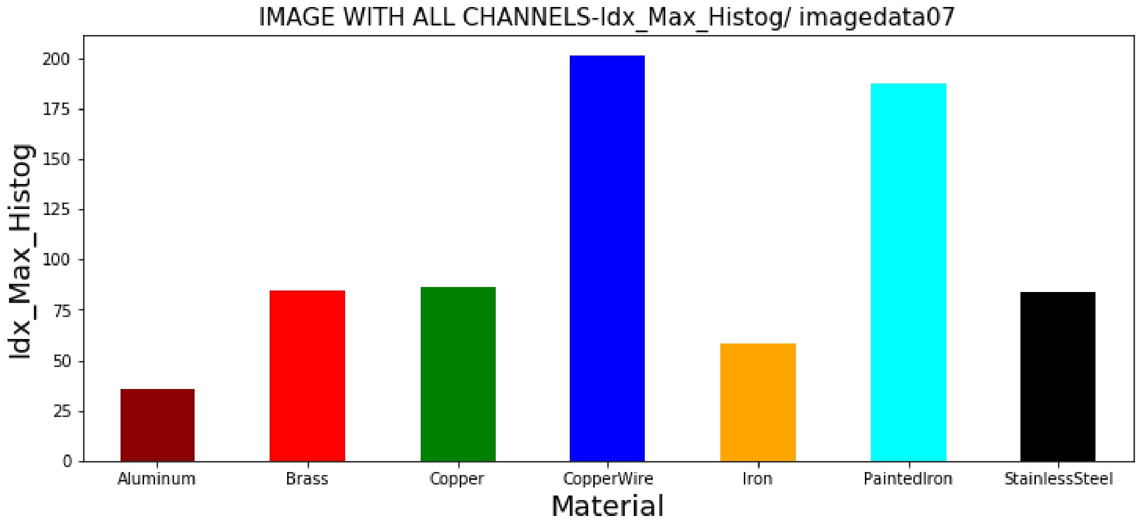
In [145]:

```
df_Max_Histog = pd.DataFrame(df_plot1.Max_Histog)
color = ['darkred','red','green','blue','orange','cyan','black','yellow']

df_Max_Histog.plot(kind='bar', y=0, color=color, legend=False, rot=0, figsize=(12,5))
plt.title('IMAGE WITH ALL CHANNELS-Max_Histog/ '+xsheet,fontsize=15)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Max_Histog',fontsize=18)
plt.savefig(xsheet+'_Max_Histog.png')   # Save  Figure
plt.show()
```



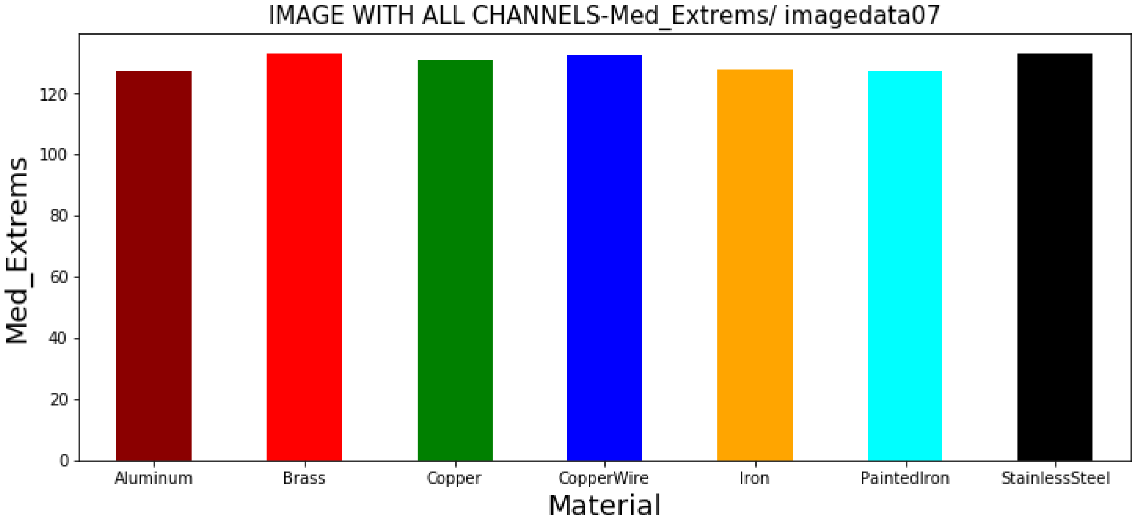
In [146]:

```
df_Idx_Max_Histog = pd.DataFrame(df_plot1.Idx_Max_Histog)
color = ['darkred','red','green','blue','orange','cyan','black','yellow']

df_Idx_Max_Histog.plot(kind='bar', y=0, color=color, legend=False, rot=0, figsize=(12,5
))
plt.title('IMAGE WITH ALL CHANNELS-Idx_Max_Histog/ '+xsheet,fontsize=15)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Idx_Max_Histog',fontsize=18)

plt.savefig(xsheet+'_Idx_Max_Histog.png')   # Save  Figure
plt.show()
```



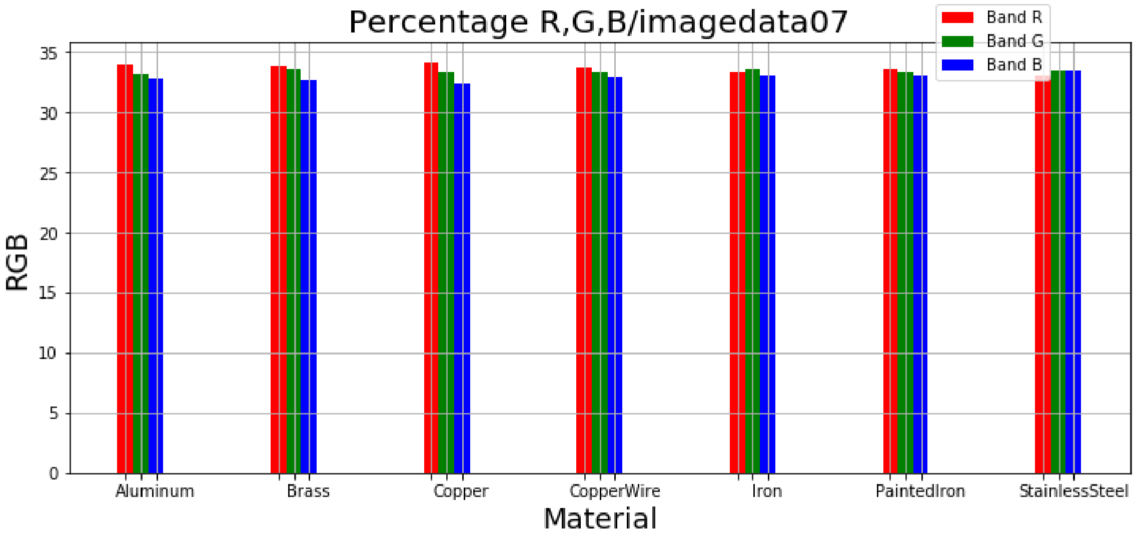
In [147]:

```
df_Med_Extrems = pd.DataFrame(df_plot1.Med_Extrems)
color = ['darkred','red','green','blue','orange','cyan','black','yellow']

df_Med_Extrems.plot(kind='bar', y=0, color=color, legend=False, rot=0, figsize=(12,5))
plt.title('IMAGE WITH ALL CHANNELS-Med_Extrems/ '+xsheet,fontsize=15)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Med_Extrems',fontsize=18)
#plt.legend(loc='upper right', ncol=3, fancybox=True, shadow=True)

plt.savefig(xsheet+'_Med_Extrems.png')   # Save  Figure
plt.show()
```



In [148]:

```
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())
xticks = df_plot1.index
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'+'/'+xsheet,fontsize=20)
plt.xlabel('Material',fontsize=18)
plt.ylabel('RGB',fontsize=18)
plt.grid(True)
plt.xticks(xtick_loc, xticks, rotation=0)
plt.legend(bbox_to_anchor=(.8,0.8),\
    bbox_transform=plt.gcf().transFigure)
plt.savefig(xsheet+"_Bar Diagram_perc_RGB.png")
plt.show()
```



In [149]:

```
# Create pivot table
df_plot2 = df.groupby('Material')['Color_dec','Med_Extrems', 'Max_Histog',
                                  'Idx_Max_Histog','Min_Histog','Idx_Min_Histog',
                                   'perc_R', 'perc_G', 'perc_B'].mean()
df_plot2
```

Out[149]:

	Color_dec	Med_Extrems	Max_Histog	Idx_Max_Histog	Min_Histog	Idx_
Material						
Aluminum	1.605632e+07	127.500000	521959.000000	35.750000	73.750000	
Brass	1.664602e+07	133.055556	453770.666667	84.666667	1.666667	
Copper	1.664606e+07	131.222222	356972.333333	86.000000	0.000000	
CopperWire	1.664602e+07	132.277778	432326.666667	201.333333	1.000000	
Iron	1.664604e+07	127.666667	454483.666667	58.333333	8.333333	
PaintedIron	1.605632e+07	127.500000	475200.000000	187.666667	173.666667	
StainlessSteel	1.605632e+07	132.777778	353889.666667	83.666667	0.000000	

In [150]:

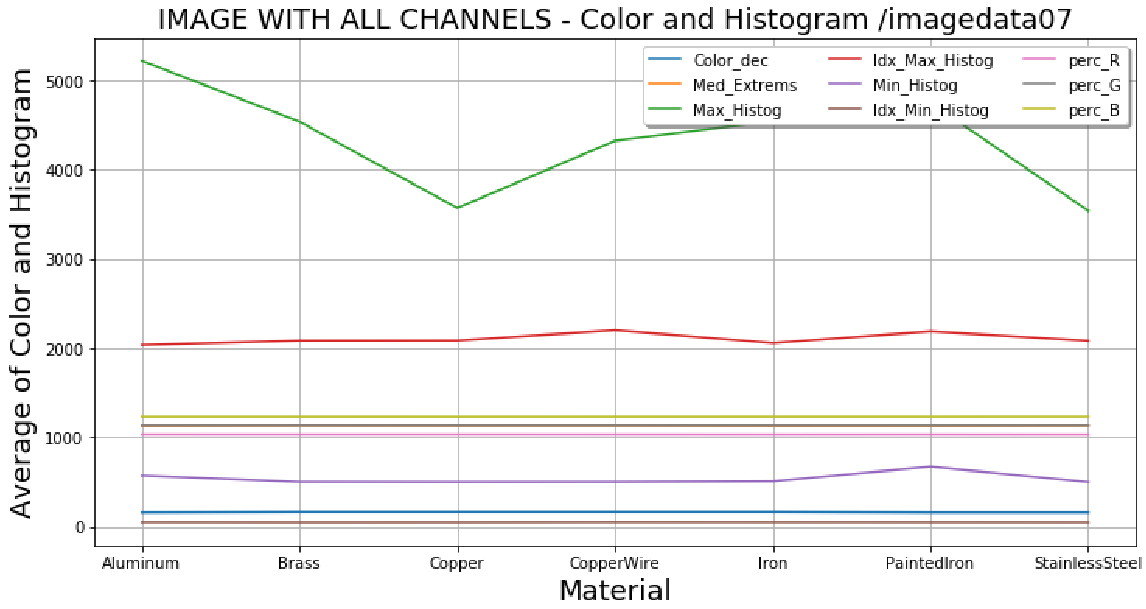
```
# Arranje values
df_plot2.Color_dec       = df_plot2.Color_dec / 100000
df_plot2.Med_Extrems     = df_plot2.Med_Extrems + 1000
df_plot2.Max_Histog      = df_plot2.Max_Histog / 100
df_plot2.Idx_Max_Histog = df_plot2.Idx_Max_Histog + 2000
df_plot2.Min_Histog      = df_plot2.Min_Histog +500
df_plot2.Idx_Min_Histog = df_plot2.Idx_Min_Histog + 50

df_plot2.perc_R = df_plot2.perc_R + 1000
df_plot2.perc_G = df_plot2.perc_G + 1100
df_plot2.perc_B = df_plot2.perc_B + 1200

df_plot2
##
```

Out[150]:

	Color_dec	Med_Extrems	Max_Histog	Idx_Max_Histog	Min_Histog	Idx_Min_
Material						
Aluminum	160.563190	1127.500000	5219.590000	2035.750000	573.750000	51.(
Brass	166.460150	1133.055556	4537.706667	2084.666667	501.666667	50.(
Copper	166.460577	1131.222222	3569.723333	2086.000000	500.000000	50.:
CopperWire	166.460150	1132.277778	4323.266667	2201.333333	501.000000	51.(
Iron	166.460363	1127.666667	4544.836667	2058.333333	508.333333	51.:
PaintedIron	160.563190	1127.500000	4752.000000	2187.666667	673.666667	51.(
StainlessSteel	160.563190	1132.777778	3538.896667	2083.666667	500.000000	50.:

In [151]:

```
loc_Array_sum = np.arange(len(df_plot1.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_plot1.index, rotation=0)
plt.title('IMAGE WITH ALL CHANNELS - Color and Histogram /'+xsheet,fontsize=18)
plt.xlabel('Material',fontsize=18)
plt.ylabel('Average of Color and Histogram',fontsize=18)
plt.legend(loc='upper right', ncol=3, fancybox=True, shadow=True)

plt.savefig(xsheet+'_Color_and_Histogram.png')   # Save  Figure
plt.show()
```



In [152]:

```
df_plot2.perc_R = df_plot2.perc_R - 1000
df_plot2.perc_G = df_plot2.perc_G - 1100
df_plot2.perc_B = df_plot2.perc_B - 1200
```

In [153]:

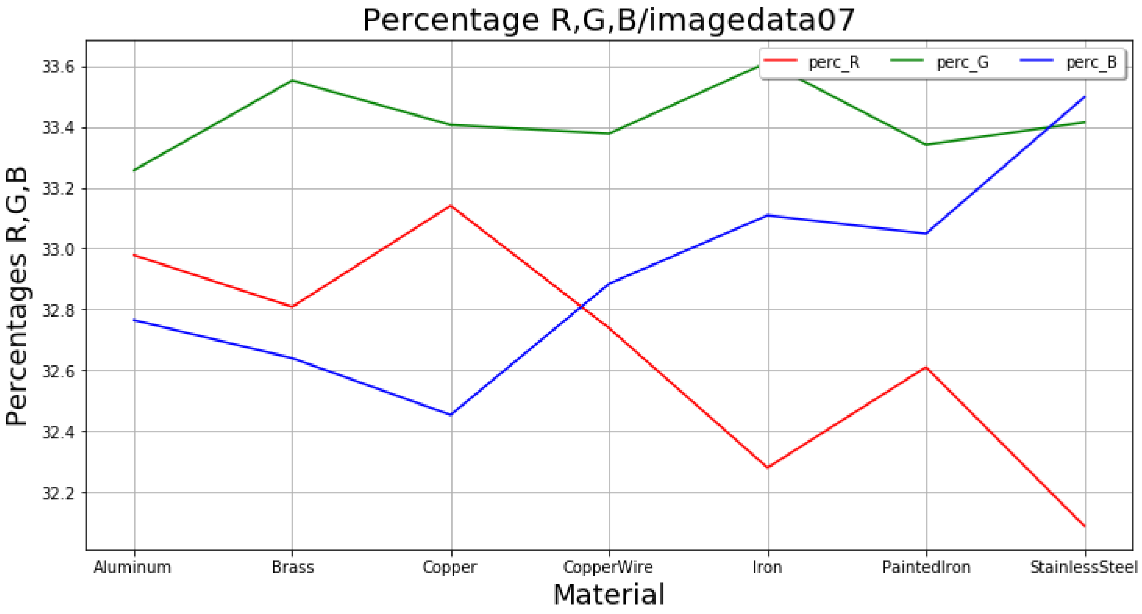
```
df_plot2.perc_R = df_plot2.perc_R - 1
df_plot2
```

Out[153]:

	Color_dec	Med_Extrems	Max_Histog	Idx_Max_Histog	Min_Histog	Idx_Min_
Material						
Aluminum	160.563190	1127.500000	5219.590000	2035.750000	573.750000	51.(
Brass	166.460150	1133.055556	4537.706667	2084.666667	501.666667	50.(
Copper	166.460577	1131.222222	3569.723333	2086.000000	500.000000	50.:
CopperWire	166.460150	1132.277778	4323.266667	2201.333333	501.000000	51.(
Iron	166.460363	1127.666667	4544.836667	2058.333333	508.333333	51.:
PaintedIron	160.563190	1127.500000	4752.000000	2187.666667	673.666667	51.(
StainlessSteel	160.563190	1132.777778	3538.896667	2083.666667	500.000000	50.:

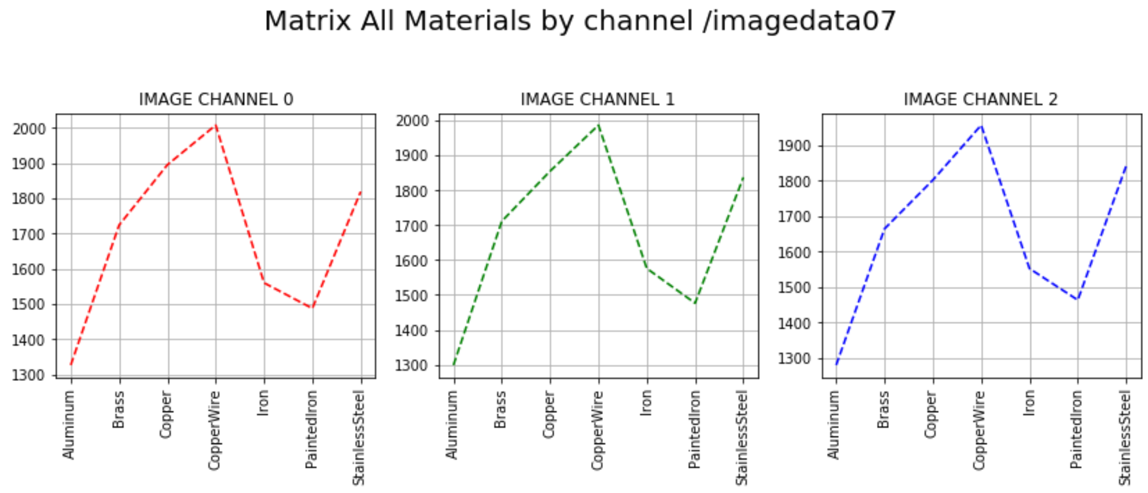
In [154]:

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

df_plot2.plot(y=["perc_R","perc_G","perc_B"],
             figsize=(12,6), grid=True, color=('r','g','b') )

#-------

#plt.plot(df_plot.Array_sum)
plt.title('Percentage R,G,B'+'/'+xsheet,fontsize=20)
plt.ylabel('Percentages R,G,B',fontsize=18)
plt.xticks(xtick_loc, df_plot1.index, rotation=0)
plt.xlabel('Material',fontsize=18)
plt.legend(loc='upper right', ncol=3, fancybox=True, shadow=True)
plt.savefig(xsheet+"_Line Graph Percentage RGB.png")
plt.show()
```



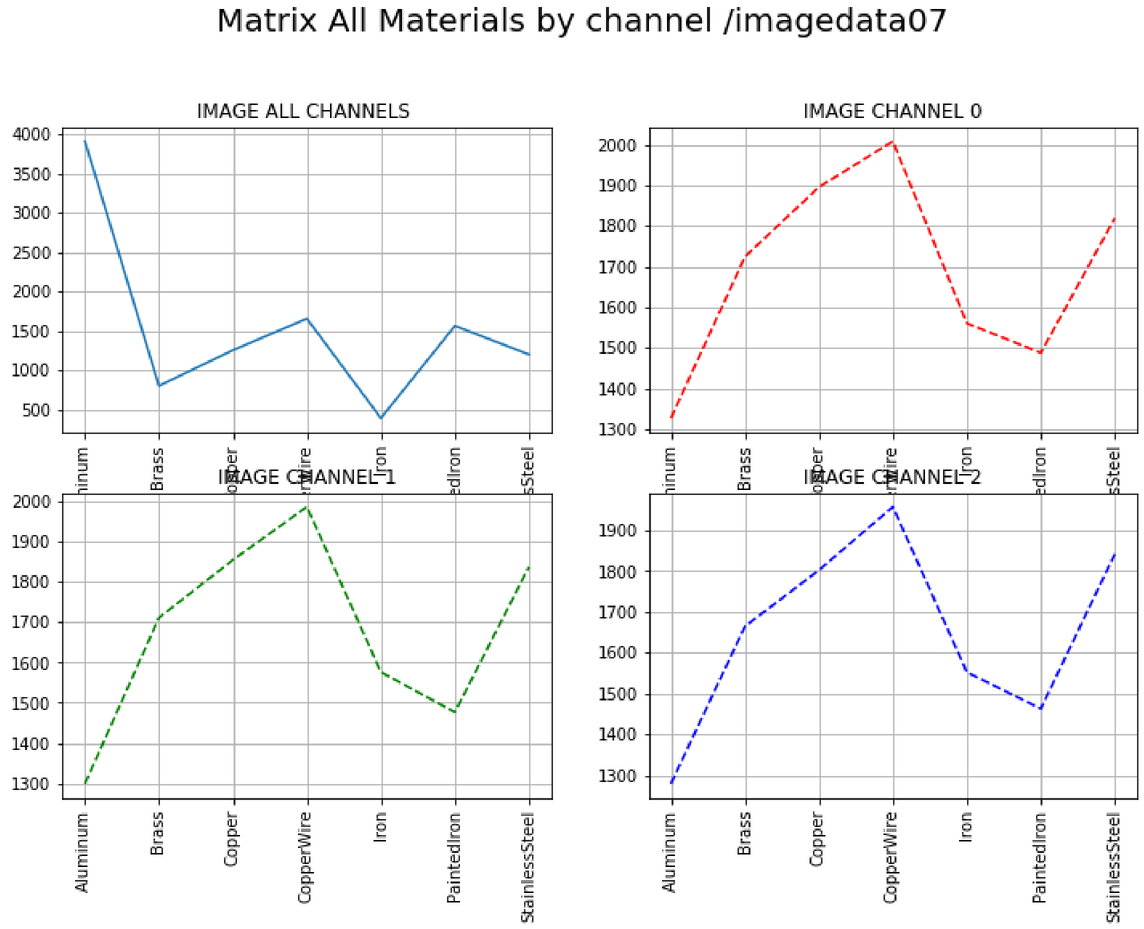
In [155]:

```
# Plot Channel separately
# Create pivot table
df_plot3 = df.groupby('Material')['All_Bands','Sum_Ch0', 'Sum_Ch1', 'Sum_Ch2'].mean()
df_plot3

import matplotlib.gridspec as gridspec

fig = plt.figure(figsize=(14,8))
gs0 = gridspec.GridSpec(2, 1)   # Spaces between graphics
fig.subplots_adjust(top=0.85)
fig.suptitle('Matrix All Materials by channel /'+xsheet,size=20) # or plt.suptitle('Mai
n title')
plt.tight_layout()

plt.subplot(231)
plt.title('IMAGE CHANNEL 0')
plt.xticks(rotation=90)
plt.grid(True)
plt.plot(df_plot3.Sum_Ch0, 'r--')

plt.subplot(232)
plt.title('IMAGE CHANNEL 1')
plt.xticks(rotation=90)
plt.grid(True)
plt.plot(df_plot3.Sum_Ch1,  'g--')

plt.subplot(233)
plt.title('IMAGE CHANNEL 2')
plt.xticks(rotation=90)
plt.plot(df_plot3.Sum_Ch2,  'b--')
plt.grid(True)
#plt.legend(loc='upper right', ncol=3, fancybox=True, shadow=True)

plt.savefig(xsheet+'_Matrix_Channels.png')   # Save  Figure
plt.show()
```



In [156]:

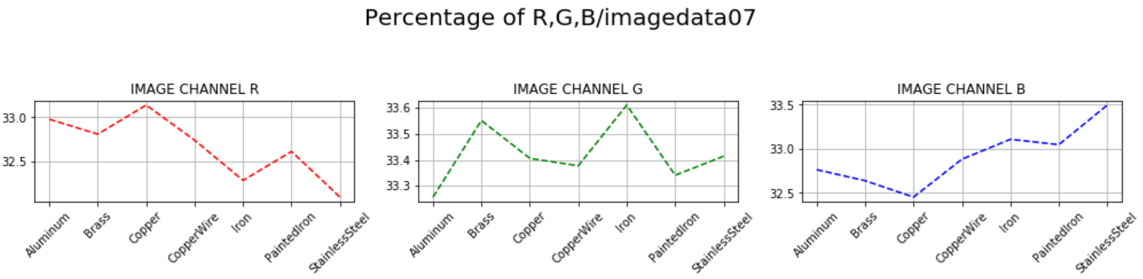
```
import matplotlib.gridspec as gridspec


fig = plt.figure(figsize=(12,8))
gs0 = gridspec.GridSpec(2, 1)   # Spaces between graphics

for i in range(1,5):
    ax=fig.add_subplot(2,2,i)

    plt.xticks(rotation=90)
    if i == 1:
        ax=plt.plot(df_plot3.All_Bands)
        plt.grid(True,which="both",ls="-")
        plt.title('IMAGE ALL CHANNELS ')
    if i == 2:
        ax=plt.plot(df_plot3.Sum_Ch0, 'r--')
        plt.grid(True,which="both",ls="-")
        plt.title('IMAGE CHANNEL 0')
    if i == 3:
        ax=plt.plot(df_plot3.Sum_Ch1, 'g--')
        plt.grid(True,which="both",ls="-")
        plt.title('IMAGE CHANNEL 1')
    if i == 4:
        ax=plt.plot(df_plot3.Sum_Ch2, 'b--')
        plt.grid(True,which="both",ls="-")
        plt.title('IMAGE CHANNEL 2')

        # Tight layout often produces nice results
# but requires the title to be spaced accordingly
fig.subplots_adjust(top=0.85)
fig.suptitle('Matrix All Materials by channel /'+xsheet,size=20) # or plt.suptitle('Mai
n title')
plt.show()
```


Matrix All Materials by channel /imagedata07
IMAGE ALL CHANNELS
500
1000
1500
2000
2500
3000
3500
4000
IMAGE CHANNEL 0
1300
1400
1500
1600
1700
1800
1900
2000
IMAGE CHANNEL 1
IMAGE CHANNEL 2
Aluminum
Brass
Copper
CopperWire
Iron
PaintedIron
StainlessSteel

In [157]:

```
# 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_plot2.perc_R, 'r--')

plt.subplot(232)
plt.title('IMAGE CHANNEL G')
plt.xticks(rotation=45)
plt.grid(True)
plt.plot(df_plot2.perc_G,  'g--')

plt.subplot(233)
plt.title('IMAGE CHANNEL B')
plt.xticks(rotation=45)
plt.plot(df_plot2.perc_B,  'b--')
plt.grid(True)

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

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



In [116]:

```
print("Finished")
```

```
Finished
```

In [ ]: