Assignment 1: Building a Better Contact Sheet

In the lectures for this week you were shown how to make a contact sheet for digital photographers, and how you can take one image and create nine different variants based on the brightness of that image. In this assignment you are going to change the colors of the image, creating variations based on a single photo. There are many complex ways to change a photograph using variations, such as changing a black and white image to either "cool" variants, which have light purple and blues in them, or "warm" variants, which have touches of yellow and may look sepia toned. In this assignment, you'll be just changing the image one color channel at a time

Your assignment is to learn how to take the stub code provided in the lecture (cleaned up below), and generate the following output image:

From the image you can see there are two parameters which are being varied for each sub-image. First, the rows are changed by color channel, where the top is the red channel, the middle is the green channel, and the bottom is the blue channel. Wait, why don't the colors look more red, green, and blue, in that order? Because the change you to be making is the ratio, or intensity, or that channel, in relationship to the other channels. We're going to use three different intensities, 0.1 (reduce the channel a lot), 0.5 (reduce the channel in half), and 0.9 (reduce the channel only a little bit).

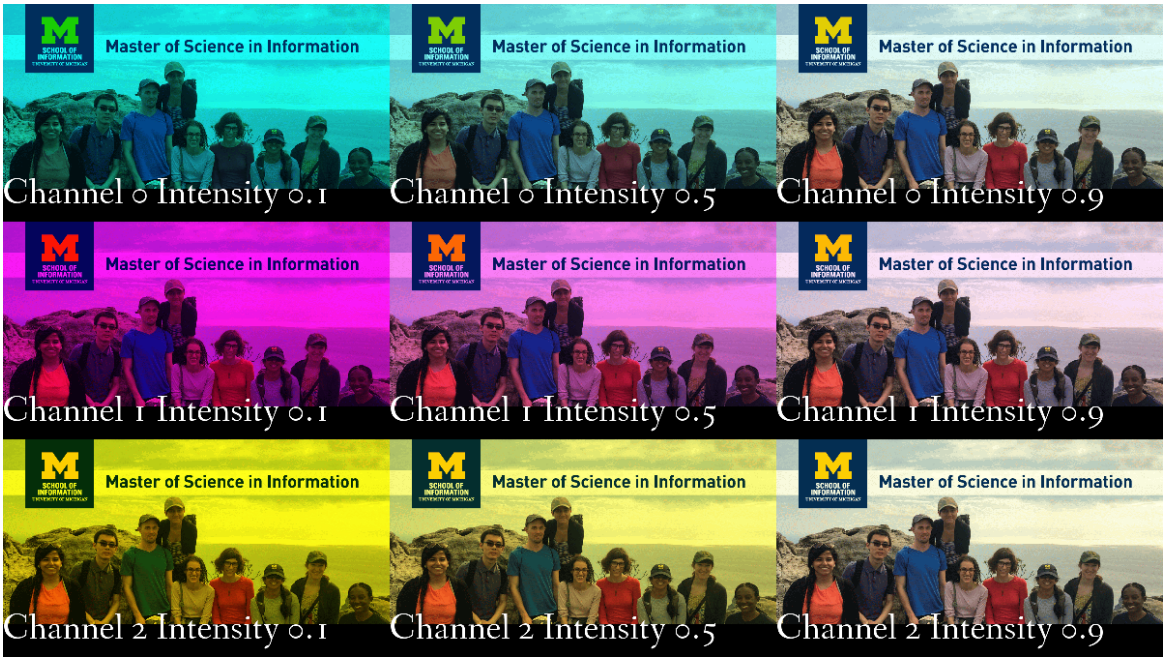
For instance, a pixel represented as (200, 100, 50) is a sort of burnt orange color. So the top row of changes would create three alternative pixels, varying the first channel (red). one at (20, 100, 50), one at (100, 100, 50), and one at (180, 100, 50). The next row would vary the second channel (blue), and would create pixels of color values (200, 10, 50), (200, 50, 50) and (200, 90, 50).

Note: A font is included for your usage if you would like! It's located in the file `readonly/fanwood-webfont.ttf`

Need some hints? Use them sparingly, see how much you can get done on your own first! The sample code given in the class has been cleaned up below, you might want to start from that.

In [86]:

```
import PIL
from PIL import Image
from PIL import ImageDraw
from PIL import ImageFont
import inspect
# read image and convert to RGB
image=Image.open("readonly/msi_recruitment.gif")
image=image.convert('RGB')

#Create the blackbar
blackbar = PIL.ImageDraw.Draw(image)
# Set the blackbar at the bottom of each figure
blackbar.rectangle([(0,int(image.height*0.85)),(image.width, image.height)], fil
l = 'black', outline = 'red')

#Font for the label text
fnt = PIL.ImageFont.truetype("readonly/fanwood-webfont.ttf",75)

# Split the image into its respective colors
images=[]
labels = []
#iterate over the intensities and channels
for i in range(3):
    for j in (0.1,0.5,0.9):
        source = image.split() #split img into its rgb values
        mid = source[i].point(lambda x:x*j)
        source[i].paste(mid) #change the required value
        im = Image.merge(image.mode, source) #incorporate the modified rgb back
 into the image
        labels.append("Channel {} Intensity {}".format(i,j))
        images.append(im) #append image to list of images.

# create a contact sheet from different color intensities
first_image=images[0]
contact_sheet=PIL.Image.new("RGB", size = (first_image.width*3,first_image.heigh
t*3))
draw = PIL.ImageDraw.Draw(contact_sheet)
x=0
y=0

for i,img in enumerate(images):
    # Lets paste the current image into the contact sheet
    contact_sheet.paste(img, (x, y) )
    draw.text((x,y+(first_image.height)*0.80), labels[i], font=fnt)
    # Now we update our X position. If it is going to be the width of the image,
then we set it to 0
    # and update Y as well to point to the next "line" of the contact sheet.
    if x+first_image.width == contact_sheet.width:
        x=0
        y=y+first_image.height
    else:
        x=x+first_image.width

# resize and display the contact sheet
contact_sheet = contact_sheet.resize((int(contact_sheet.width/2),int(contact_she
et.height/2) ))
display(contact_sheet)
print(labels)
```




```
['Channel 0 Intensity 0.1', 'Channel 0 Intensity 0.5', 'Channel 0 In
tensity 0.9', 'Channel 1 Intensity 0.1', 'Channel 1 Intensity 0.5',
'Channel 1 Intensity 0.9', 'Channel 2 Intensity 0.1', 'Channel 2 Int
ensity 0.5', 'Channel 2 Intensity 0.9']
```

HINT 1

Check out the `PIL.ImageDraw module` for helpful functions

HINT 2

Did you find the `text()` function of `PIL.ImageDraw` ?

HINT 3

Have you seen the `PIL.ImageFont` module? Try loading the font with a size of 75 or so.

HINT 4

These hints aren't really enough, we should probably generate some more.