Activité 1 sur le thème 4 : généralités sur la photo numérique-modifier une photo numérique à l'aide de Python

Consignes :

Enregistrer immédiatement le fichier word dans le dossier approprié (snt/theme4).
 Compléter ce document Word . Il faudra soigner la rédaction (phrases complètes). A la fin de l'heure , enregistrer votre travail sur clé usb et déposer le document dans l'espace élève de Pronote. (page d'accueil ou cahier de texte – onglet déposer ma copie)

Pour commencer une vidéo interactive : lienmini.fr/3389-501

Tester ses connaissances : lienmini.fr/3389-502 p102-103

Faire une capture d'écran de ses résultats (touche imprécran) et la coller ci-dessous.

Repères historiques lienmini.fr/3389-503

Définition d'un pixel :ne image numérique est constituée d'un ensemble de points appelés pixels. Un pixel (abrévation de PICture ELement) est défini comme le plus petit élément constitutif d'une image numérique.

1ère partie : généralités : repérage et codage RVB d'un pixel

Un pixel est défini par ses coordonnées. L'origine du repère qui permet de définir ces coordonnées se trouve en haut à gauche de l'image. L'axe des abscisses est orienté de gauche à droite et l'axe des ordonnées est orienté de haut en bas.



A chaque pixel , est associé 3 nombres entiers compris entre 0 et 255 (un octet) r,v,b. Ces nombres indiquent l'intensité des couleurs ROUGE , VERTE et BLEUE.

Quelle couleur obtient on lorsque r=255 , v=255 et b=255 ? Quelle couleur obtient on lorsque r=0 , v=0 et b=0 ? Quelle couleur obtient on lorsque r=255 , v=0 et b=255 ? Quelle couleur obtient on lorsque r=255 , v=255 et b=0 ? Ouvrir le lien : https://www.geogebra.org/m/XrXT8DwS

Déplacez les cercles et voyez l'intersection entre chacun pour voir la composition des couleurs. Modifiez la valeur de chacune des couleurs à l'aide des curseurs et vérifier les couleurs obtenues aux questions précédentes.



Négatif d'une couleur :

Le négatif d'une couleur est la couleur obtenue lorsque l'on soustrait 255 à la quantité de rouge , de vert et de bleu.

Donner le négatif du rouge absolu (r=255, v=0, b=0).

Donner le négatif du vert absolu (r=0, v=255, b=0).

.....

Donner le négatif du bleu absolu (r=0, v=0, b=255).

.....



Nuance de gris :

Le gris est obtenu lorsque les quantités de rouge, de vert et de bleu sont égales.

Combien y a-t-il de nuances de gris ?

.....

Des algorithmes sont utilisés dans les appareils de photographie numérique lors de la prise de vue et lors du développement de l'image. Une image étant créée, on peut la traiter à l'aide de

programmes informatiques pour la transformer en agissant sur les trois composantes de chaque pixel.

Dans la suite, on propose de faire des activités qui utilisent des morceaux de codes en Python. On utilisera la bibiothèque PIL d'Edupython.

Un pixel est défini par ses coordonnées. L'origine du repère qui permet de définir ces coordonnées se trouve en haut à gauche de l'image. L'axe des abscisses est orienté de gauche à droite et l'axe des ordonnées est orienté de haut en bas.

Fonctions utiles :

Voici des exemples d'instructions et de morceaux de codes qui permettent de travailler sur une image.

from PIL import Image	#On importe le module Image de la bibliothèque PIL
<pre>img1=Image.open("Mon_Image.jpg")</pre>	# Ouverture du fichier Mon_Image.jpg et création de l'objet img1
<pre>img1.getpixel((i, j))</pre>	# On récupère la couleur du pixel (i,j) de l'image imgl sous la forme d'un tuple (r, g, b)
<pre>img.putpixel((i, j), (r,v,b))</pre>	# On attribue la couleur (r,v,b) au pixel (i,j)
<pre>img.save("img2.jpg")</pre>	# on enregistre l'image crée sous le sous nom img2.jpg
img2.show()	#On montre l'image img2

IMPORTANT : avant de commencer, il faut créer un dossier (si ce n'est pas déjà fait) « theme4 » dans lequel on enregistrera tous les fichiers

- Le document word activite1theme4
- L'image que vous souhaitez modifier
- Le module python

2^{ème} partie : passage au négatif – nuance de gris-permutation des couleurs

1.Ouvrir l'image « joconde.jpg » (<u>mathssa.fr/joconde</u>) et enregistrer cette image dans le dossier « theme4 ».

Ouvrir un nouveau module d'Edupython (nouveau-nouveau module python et fermer d'éventuels modules parasites)et l'enregistrer dans le dossier « theme4 » sous le nom « negatif.py »

Le programme ci-dessous permet de récupérer pour chaque pixel le triplet (r,v,b) puis on modifie ce triplet afin d'obtenir l'image en « négatif » (le négatif d'une image s'obtient en faisant soustrayant 255 à chacun des nombres r,v,b)

Recopier et compléter le programme ci-dessous : (inutile de recopier les commentaires) On pourra également récupérer le programme à l'adresse : <u>mathssa.fr/negatif.py</u>)

Joindre une capture d'écran de l'image négatif.jpg

Si vous avez une image, disponible vous pouvez obtenir le négatif de votre image, en utilisant le programme précédent (remplacer « joconde.jpg » et « négatif.jpg » - ne pas oublier d'enregistrer votre image) 2.Ouvrir un nouveau module d'Edupython et l'enregistrer dans le dossier « theme4 » sous le nom « nuancegris.py »

En s'inspirant du programme précédent, écrire un programme permettant d'obtenir une image de la Joconde en nuance de gris. On sauvegardera l'image sous le nom « nuancegris.jpg)

3.Ouvrir un nouveau module d'Edupython et l'enregistrer dans le dossier « theme4 » sous le nom « permutation.py »

Le programme ci-dessous permet d'effectuer pour chaque pixel une permutation du triplet (r,v,b) .

Recopier et compléter le programme ci-dessous : (inutile de recopier les commentaires). On pourra également récupérer le programme à l'adresse : <u>mathssa.fr/permutation.py</u>) from PIL import Image #Importe le module Image de la bibliothă⁻que PIL

```
image0 = Image.open("joconde.jpg") # Ouverture du fichier image source et crÃ@ation de l'objet Image
dimx = image0.size[0] # Largeur de l'image
dimy = image0.size[1] # hauteur de l'image
image1 = Image.new('RGB', (dimx, dimy), (0, 0, 0)) # CrÃ@ation d'une nouvelle image ayant la meme dimension
for i in range(dimy):
    for j in range(dimy):
        r, v, b = image0.getpixel((i, j)) # On rÃ@cupÃ"re la couleur du pixel (i,j) sous la forme d'un triplet (r, v, b)
        image1.save("permutation.jpg") # on sauvegarde sous un autre nom
    image1.show()
```

Joindre une capture d'écran de l'image permutation.jpg.

3ème partie : mosaïque

Dans cette parttie, on se propose de dessiner une mosaïque inspirée par le portrait de Marylin Monroe réalisé par Andy Warhol.

Pour cela, on utilisera une image en couleur de votre choix ou l'image dont le lien est mathssa.fr/Femme.jpg et on copiera cette image dans un dossier « photo numerique » . Le nouveau module python s'appellera mosaique.py.



On souhaite coder un programme qui reproduira l'image ci-dessous avec les traitements proposés sous l'image sur sa table de couleurs (R1,V1,B1).







Traitements

- Image n° 2 : les nouvelles couleurs (R2,V2,B2) des pixels sont permutées de telle manière que : R2=B1, V2=R1, B2=V1
- Image n° 3 : les nouvelles couleurs (R3,V3,B3) des pixels sont permutées de telle manière que : R3=V1, V3=B1, B3=R1
- Image n° 4 : les nouvelles couleurs (R4,V4,B4) des pixels sont inversées de telle manière que : R4=255-R1, V4=255-V1, B4=255-B1
- Image n° 5 : elle représente l'image en niveaux de gris en faisant la moyenne des composantes (R1,V1,B1) de l'image initiale
- Image n° 6 : les couleurs (R5,V5,B5) des pixels de l'image n° 5 sont inversées de telle manière que : R6=255-R5, V6=255-V5, B6=255-B5
- Image n° 7 : elle représente l'image initiale en supprimant la composante bleue.
- Image n° 8 : elle représente l'image initiale en supprimant la composante rouge.

- Image n° 9 : elle représente l'image initiale en supprimant la composante verte.
- Image n° 10 : elle représente l'image initiale en ne gardant que la composante rouge.
- Image n° 11 : elle représente l'image initiale en ne gardant que la composante verte.
- Image n° 12 : elle représente l'image initiale en ne gardant que la composante bleue.

Pour cela on complètera le code suivant : (disponible à l'adresse <u>mathssa.fr/mosaique.py</u>)

from PIL import Image #Importe le module Image de la bibliothèque PIL		
<pre>image0 = Image.open("Femme.jpg") # Ouverture du fichier image source et création dimx = image0.size[0] # largeur de l'image dimy = image0.size[1] # hauteur de l'image</pre>	de l'objet Image	
<pre>dimx = image0.size[0] # largeur de l'image dimy = image0.size[1] # hauteur de l'image image1 = Image.new('RGB', (3*dimx, 4*dimy), (0, 0, 0)) # Création d'une nouvelle for i in range(dimx): # parcours des deux dimensions pour modifier chaque pixe for j in range(dimy):</pre>	image 3 fois plus large et 4 fois plus haute l	
<pre>r, v, b = image0.getpixel((i, j)) # On récupère la couleur du pixel (i, j) sous la forme d'un tuple (r, g, b) image1.putpixel((i, j), (r,v,b)) # On recopie cette image dans le colin en haut à gauche image1.putpixel((dimxti, j), (,)) #Etape 2 : permutation de (r,v,b) image1.putpixel((2*dimxti, j), (,)) #Etape 3 : permutation de (r,v,b) image1.putpixel((i, dimytj), (,)) #Etape 4 : passage au négatif de l'image #Etape 5 : olle perpécate l'image on prioratur de apris en friedrat la peuperate des composates (p, v, b) de l'image initiale</pre>		
<pre>g=int((r+v+b)/3) image1.putpixel((,), (.,.,.)) #Etape 6 : passage au négatif de l' image1.putpixel((,), (.,.,.)) # Etape 7 : image initiale en supp image1.putpixel((,), (.,.,.)) # Etape 8 :image initiale en supp image1.putpixel((,), (.,.,.)) # Etape 9 :image initiale en supp image1.putpixel((,), (.,.,.)) # Etape 10 :image initiale en supp image1.putpixel((dimx+i, 3*dimy+j), (.,.,.)) #Etape 11 :image initiale en image1.putpixel((2*dimx+i, 3*dimy+j), (.,.,.)) #Etape 12 :image initiale</pre>	image de l'étape 5 primant la composante bleue. mant la composante rouge imant la composante verte primant la composante bleue et verte. supprimant la composante rouge et bleue en supprimant la composante rouge et verte	
<pre>image1.save("mosaique.jpg") # on sauvegarde sous un autre nom image1.show()</pre>	Console Python >>> *** Console de processus distant Réinitialise File "C:\Users\ORDI\OneDrive - ac-clemont. professions])mathe 2021 2015UT at ALGONITHMY	

Joindre ci-dessous une capture d'écran de votre travail. Tester ce programme avec vos propres images.