**Baboumian Patrick**

Examen de télédétection (Python avancé) - 2018-2019

Les réponses du QCM sont à indiquer en **noircissant** à l'encre noire ou bleue ($\square \rightarrow \blacksquare$) les cases correspondantes à vos réponses. Il n'y a qu'une seule bonne réponse par question:

- bonne réponse : +1
- pas de réponse : 0
- mauvaise réponse ou plusieurs réponses apportées : -0.5

Pour décocher une case cochée par erreur, prendre bien soin d'effacer le contenu de cette case (ne pas entourer une autre réponse, etc.) sinon celle-ci sera considérée comme cochée.

Information

- Les deux images sont à télécharger : karasiak.net/rosamond.zip.
 - Veuillez rendre en fin de cours en plus de ce document, un dossier zippé à votre nom contenant les fichiers demandés (à la fin des questions sont spécifiés les **fichiers à rendre**).
- Le dossier est à rendre de préférence en l'uploadant via l'adresse karasiak.net/upload.php.

1 QCM /6

Question 1 Que retourne la commande python `gdal.Open()` si on spécifie en premier paramètre le chemin d'une image existante ?

- ☐ Un tableau numpy ☐ Un objet GDAL. ☐ Une image. ☐ La valeur *None*.

Question 2 Un tableau numpy a été initialisé comme suit : `M = np.zeros((250,250,4))`. Que retourne la commande `M.size` ?

- ☐ 3 ☐ (250,250,4) ☐ Une erreur ☐ 250000

Question 3 Pourquoi est-il déconseillé de charger directement l'image avec la commande suivante : `gdal.Open('rosamond_S2.tif').ReadAsArray()` ?

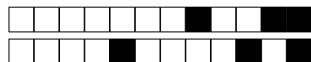
- ☐ Car le format du tableau ne sera pas au format X,Y,Z (nb de lignes, nb de colonnes, nb de bandes spectrales).
- ☐ Parce qu'on ne peut pas enregistrer le résultat.
- ☐ Car `gdal` n'est pas optimisé pour les fichiers geotiff.
- ☐ Car la fonction `readAsArray()` ne retourne qu'une dimension à la fois.

Question 4

65	122	2
70	2	255
76	82	77

La valeur au format entier du filtre *median* pour le voisinage défini dans le tableau ci-dessus (carré de 3x3) est :

- ☐ 255 ☐ 76 ☐ 77 ☐ 2



+19/2/5+

Question 5

196	31	256
119	172	139
124	16	65

Si le tableau se nomme M, comment puis-je accéder à la valeur 256 ?

- ☐ M[3,1]
 ☐ M[1,3]
 ☐ M[0,2]
 ☐ np.where(M=256)

Question 6 Quelle est la dimension de l'image `rosamond_crops.png` (hauteur, largeur, bandes) ?

- ☐ 528, 533,4
 ☐ 528,533,3
 ☐ 4,528,533
 ☐ 533, 528,4

2 Programmation Python /14

2.1 Manipulation d'une image /6

À Rosamond aux États-Unis, afin de pouvoir cultiver malgré la sécheresse, plusieurs procédés d'irrigation sont utilisés. Les parcelles sont de taille carrée ce qui nous arrange bien pour traiter les images ! Pour faciliter les traitements par la suite, nous supposons que l'image se divise en 4 par 4 (comme un damier).



Figure 19: `rosamond_crops.png`

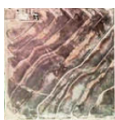
Question 7 Depuis python, ouvrez avec `gdal` l'image `rosamond_crops.png`. et extrayez de manière automatique la parcelle qui est irriguée de manière circulaire (il s'agit du carré de la deuxième ligne et deuxième colonne, exemple Fig 1). Sauvegardez le résultat sous `rosamond_crops_circle.npy` avec la fonction `save()` de numpy.



Fichiers à rendre : `rosamond_crops_circle.npy`.

..... ☐0 ☐1 ☐2 Cadre réservé au correcteur

Question 8 Faites la même chose avec le carré pour la dernière colonne de la ligne numéro 2. Sauvegardez le résultat sous `rosamond_crops_curves.npy` avec la fonction `np.save()`.



Fichiers à rendre : `qu7_8.py` (contient le code servant à générer les deux carrés), `rosamond_crops_curves.npy`.

..... ☐0 ☐1 ☐2 Cadre réservé au correcteur

Question 9 Créez une fonction qui prend en entrée le tableau en entier de l'image, et qui enregistrera automatiquement chaque carré de l'image `rosamond_crops.png`. Sauvegardez ces 16 résultats (4*4) sous la forme `numéroDuCarré.npy`.

Fichiers à rendre : `qu_9.py`, et 16 fichiers `npy`.

..... ☐0 ☐1 ☐2 Cadre réservé au correcteur



2.2 Indice EVI2 et masque de végétation /8

L'indice *EVI2* (Enhanced Vegetation Index 2, ou Indice de Végétation Amélioré 2) se distingue du NDVI par sa capacité à corriger des effets atmosphériques, Il est donc préféré par certains chercheurs. Mettez en place une fonction qui calculera l'indice *EVI 2* à partir d'une image multispectrale et qui enregistrera l'indice dans le même dossier que l'image d'origine, mais avec le suffixe *_evi2* et au format GeoTIFF *.tif*, soit *rosamond_S2_evi2.tif*. La formule est la suivante :

$$EVI2 = \frac{2.5 * (IR - R)}{IR + 2.4 * R + 1}$$

où *IR* et *R* sont les valeurs de chaque pixel respectivement dans la bande infra-rouge et rouge.

Contrairement à l'image utilisée précédemment (qui était extraite de Google Earth), vous travaillerez avec une image Sentinel-2 prise le 13 juillet 2018 (*rosamond_S2.tif*). L'activité de la végétation n'est donc plus la même et vous disposez dorénavant d'une bande supplémentaire : l'infra-rouge. Pour information, les bandes sont ordonnées comme suit : bleu, vert, rouge, infra-rouge.

N'oubliez pas de commenter vos fonctions comme il a été vu pendant le cours (vous pouvez vous inspirer des fonctions de numpy).

2.2.1 Calcul de l'indice EVI2 /4

Question 10 Vous programmerez une fonction nommée `computeEVI2` qui calcule l'indice *EVI2* et qui prend comme arguments en entrée :

1. Le tableau contenant l'ensemble des bandes (et non le chemin de l'image),
2. Le numéro de la bande *infra-rouge*,
3. Le numéro de la bande *rouge*.

Une fois le calcul effectué, la fonction retourne un tableau contenant uniquement les valeurs de l'indice *EVI2*. Enregistrez ensuite l'indice (au format *.tif*. ex : *rosamond_S2_evi2.tif*). De préférence, enregistrez au format entier (il faudra multiplier les résultats par 100 et bien vérifier que le format d'enregistrement est de type entier (int ou byte, à vous de choisir)).

À noter, votre code doit pouvoir s'exécuter depuis le fichier python qui calcule l'indice et enregistre votre image, mais je dois pouvoir importer ce même fichier sans que la moindre ligne de code ne s'exécute, et ce afin que je puisse importer votre fichier et tester votre fonction qui calcule l'indice *EVI2*.

Fichiers à rendre : `qu_10.py`.

..... ☐0 ☐1 ☐2 ☐3 ☐4 Cadre réservé au correcteur

Connaître la valeur *EVI2* d'un pixel c'est bien, mais avoir un masque de la végétation c'est encore mieux !

2.2.2 Calcul de masque de végétation /4

Question 11 À partir de l'indice *EVI2*, trouvez un seuil adéquate afin de ne garder que les valeurs qui, pour vous, correspondent à une activité en chlorophylle des plantes.

Sauvegarder le résultat sous un format *.tif* où seules les valeurs supérieures au seuil que vous avez défini seront visibles (les autres seront classées en no data).

Fichiers à rendre : `qu_11.py`.

..... ☐0 ☐1 ☐2 ☐3 ☐4 Cadre réservé au correcteur