

Machine Learning TP6: aralSea

Melvin DUBEE - Tanguy ROUDAUT

FIPASE 24

28 novembre 2023

1 Objectif

Le but de ce TP est d'estimer la diminution de la surface de la mer d'Aral. Pour se faire, nous mettrons en oeuvre des scripts basé sur des méthodes du machine learning. On cherchera à déterminer la surface de la mer d'Aral au cours du temps en se basant sur des images satellitaires.

La procédure utilisée suivra le shcéma suivant :

1. Prétraitements
2. Extraction des descripteurs
3. Apprentissage d'un modèle de classement
4. Classement des pixels

2 Examen des données, prétraitements et extraction des descripteurs

Question 1 : Preprocessing Function

```
1 def preprocessing():
2     img73, img87 = loadImages()
3     plt.subplot(1, 2, 1)
4     plt.imshow(img73)
5     plt.title('Image 1973')
6     plt.subplot(1, 2, 2)
7     plt.imshow(img87)
8     plt.title('Image 1987')
9     plt.show()
10
11     return 0, img73, img87
```

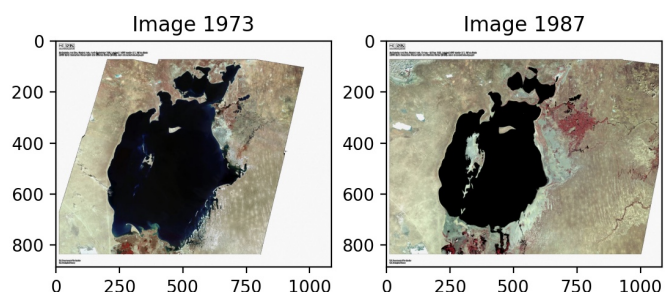


FIGURE 1 – Images chargées

Question 2 : Analyse des images

L'analyse de ces images indique que les dimensions de ces images sont les mêmes. Les valeurs des pixels sont codées sur trois canaux, en RGB. Au niveau visuel, on distingue nettement la différence de surface entre 1973 et 1987, notamment l'île située à l'ouest de la mer d'Aral qui a été multiplié au moins par dix.

```

1 featLearn, img73, img87 = preprocessing()
2 height, width, channels = img73.shape
3 print(f"Image img73 dimensions: Height = {height}, Width = {width}, Channels = {channels}")
4 height, width, channels = img87.shape
5 print(f"Image img87 dimensions: Height = {height}, Width = {width}, Channels = {channels}")
6 """return
7 Image img73 dimensions: Height = 889, Width = 1086, Channels = 3
8 Image img87 dimensions: Height = 889, Width = 1086, Channels = 3
9 """

```

Listing 1 – Main Function

Question 3 : Dimension de l'espace des descripteurs

L'image étant codée en RGB, cela veut dire que chaque pixel est décrit par trois valeurs qui correspondent à l'intensité du Rouge, Vert et Bleu. La dimension de l'espace des descripteurs est donc de trois, on a un espace tridimensionnel.

Question 4 : Images tronquées

```

1 plt.subplot(1, 2, 1)
2 plt.imshow(img73)
3 plt.title('Cropped Image 1973')
4 plt.subplot(1, 2, 2)
5 plt.imshow(img87)
6 plt.title('Cropped Image 1987')
7 plt.show()

```

Listing 2 – Main Function

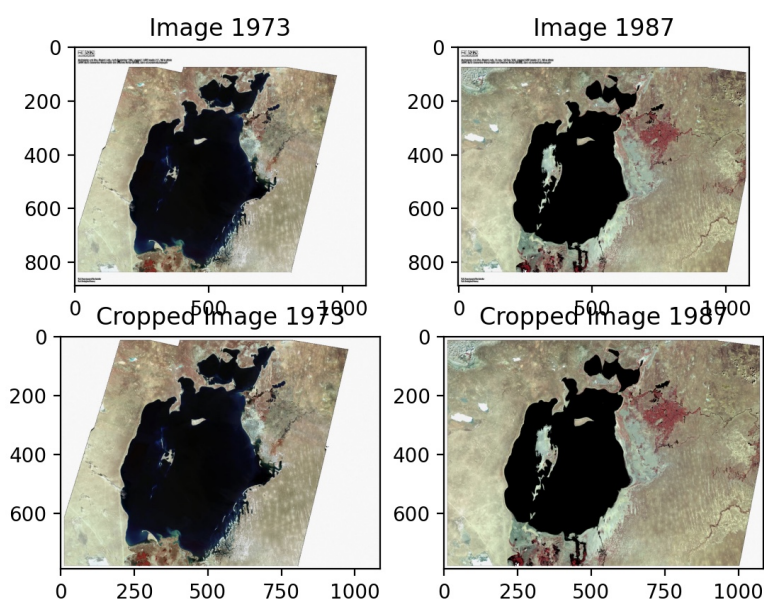


FIGURE 2 – Images tronquées

3 Constitution d'un ensemble de données d'apprentissage

Question 1 : Sous-échantillonnage

```

1 def preprocessing():
2     # [...]
3     feat, nbPix, nbFeat = selectFeatureVectors(img73, 500)
4     print(f"Number of training data: {nbPix}")
5
6     return feat, img73, img87
7     """return
8     Number of training data: 1714
9     """

```

Listing 3 – Main Function

Le résultat obtenu est le nombre de données d'apprentissage, il est donc de 1714. En sous-échantillonnant les données, on va réduire la taille de l'ensemble d'apprentissage, ce qui va nous simplifier la tâche pour la suite.

4 Première analyse des données

Question 1 : Visualisation des données en 2D et 3D

```

1 def preprocessing():
2     # [...]
3     displayFeatures2d(feat)
4     displayFeatures3d(feat)
5
6     return feat, img73, img87

```

Listing 4 – Visualisation Function

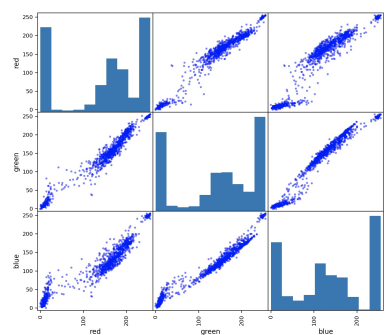


FIGURE 3 – Image 2D

Nuage de points et histogramme

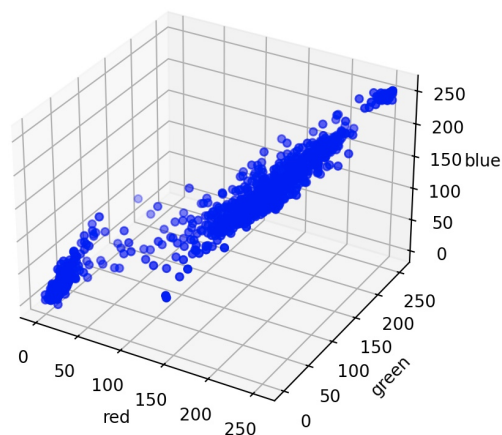


FIGURE 4 – Image 3D

Question 2 : Descriptions des graphiques

Sur le premier graphique 2D, on distingue 9 sous-graphiques combinaison des trois couleurs RGB. Ce qui se retrouve sur chacun des sous-graphiques c'est la distinction de trois clusters. En effet, on a trois niveaux d'intensités différents, un très bas, ce qui correspond à la couleur la plus foncée, la mer, un moyen, ce qui correspond à la couleur de la terre sur les extérieurs de l'image et un haut, ce qui correspondant à la terre qui fait le contour de la mer.

Sur le deuxième graphique 3D, on peut faire la même analyse mais cette fois-ci, on peut dire que l'on regroupe tous les sous-graphiques précédents, ce qui nous permet de distinguer clairement chacun des clusters. Les clusters sont donc la mer, la terre et le rivage.

5 Approche non supervisée par la méthode des k-means

Question 1 : Méthode des centres mobiles

L'algorithme des k-moyennes permet de créer des sous-ensembles de données à partir d'un ensemble de base. Pour ce TP, cet algorithme serait utile pour diviser en clusters les pixels de l'image.

Question 2 : Apprentissage non supervisée du classifieur

```
1 def unsupervisedTraining(featsLearn, method='kmeans'):  
2     answer = input('nombre de classes:')  
3     nbCluster = int(answer)  
4     if method == 'kmeans':  
5         model = KMeans(n_clusters=nbCluster, random_state=42)  
6         model.fit(featsLearn)  
7     elif method == 'gmm':  
8         ...  
9  
10    return model
```

Listing 5 – UnsupervisedTraining Function

Cette fonction est utilisée pour définir le nombre de cluster que l'on veut. Dans notre cas, comme on a pu le voir dans la partie précédente, on distingue plus ou moins trois clusters de couleur. On indiquera donc que l'on veut trois clusters.

Les paramètres importants sont le nombre de clusters et les caractéristiques d'entrées dans feat-Learn.

Question 3 : Prédiction des labels

```
1 # Apprentissage de la fonction de classement  
2 model = unsupervisedTraining(featsLearn, method='kmeans')  
3  
4 # prediction des labels sur la base d'apprentissage  
5 labels_predict = model.predict(featsLearn)
```

Listing 6 – aralsea Main Function

Question 4 : Visualisation des descripteurs d'apprentissage et de leur classe

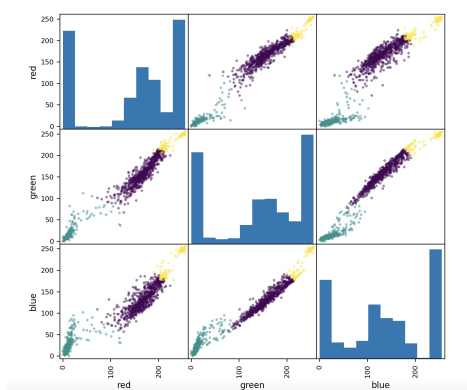


FIGURE 5 – Image 2D

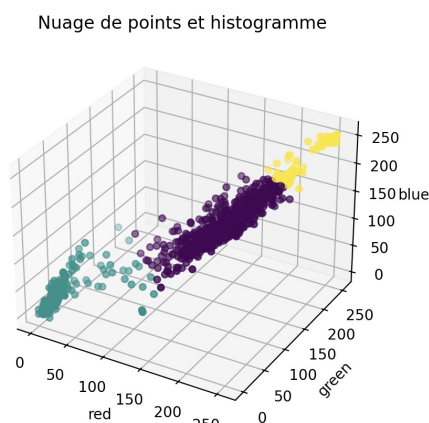


FIGURE 6 – Image 3D

On retrouve sur ces deux graphiques ce que l'on avait prédit à la question.2, on distingue bien trois clusters de couleurs différents ce qui confirme ce que l'on pensait.

Question 5 : Classification des deux images

```
1 def unsupervisedClassifying(model, feat):
2     height, width, channels = feat.shape
3     feat_matrix = feat.reshape((height * width, channels))
4     label = model.predict(feat_matrix)
5
6     return label
```

Listing 7 – unsupervisedClassifying Function

```
1 # Classifying / Predicting / Testing
2 labels_prediction_img73 = unsupervisedClassifying(model, img73)
3 labels_prediction_img87 = unsupervisedClassifying(model, img87)
```

Listing 8 – aralsea Function

Question 6 : Visualisation de la classification des deux images

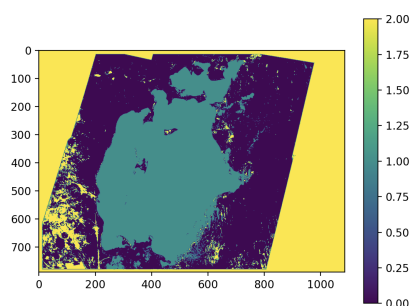


FIGURE 7 – Image 1973 Predicted Labels

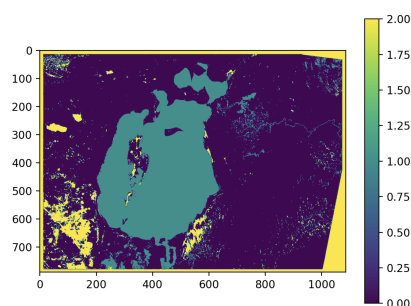


FIGURE 8 – Image 1987 Predicted Labels

Sur ces deux graphiques, on voit très bien la distinction sur l'image des trois clusters. Sur les deux images, la mer d'Aral est visible en vert/cyan, le rivage est visible en très foncé bleu/violet et l'extérieur des images en jaune.

Question 7 : Estimation de la surface et de son évolution

```

1  # Pourcentage de la mer
2  proportion_mer_img73 = (np.sum(img73_label == 1) / img73_label.size)*100
3  proportion_mer_img87 = (np.sum(img87_label == 1) / img87_label.size)*100
4
5  print("Proportion de la mer dans l'Image 1973:", proportion_mer_img73)
6  print("Proportion de la mer dans l'Image 1987:", proportion_mer_img87)
7  print("Une chute du niveau de la mer de:", proportion_mer_img73-proportion_mer_img87)
8  """return
9  Proportion de la mer dans l'Image 1973: 27.632828930016085
10 Proportion de la mer dans l'Image 1987: 21.373769627030974
11 Une chute du niveau de la mer de: 6.259059302985111
12 """

```

Listing 9 – SurfaceEvolution Function

D'après les résultats obtenus, on a un pourcentage de la mer de 27.63% en 1973 contre 21.37% en 1987. L'évolution de la surface de la mer d'Aral en 14 ans est donc d'approximativement 6.26%.

Question 8 : Estimation de la surface et de son évolution

k=6

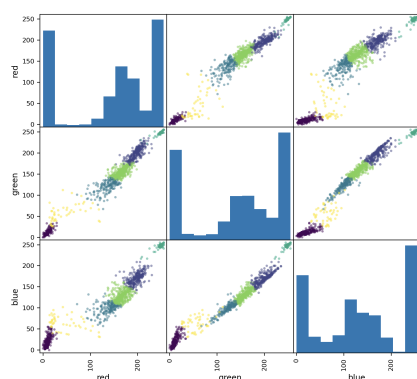


FIGURE 9 – Image 2D

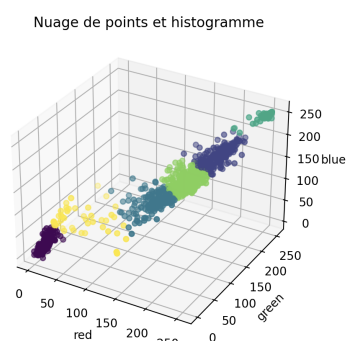


FIGURE 10 – Image 3D

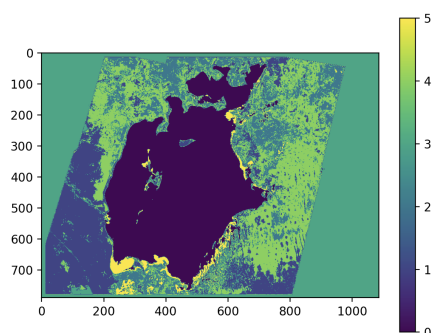


FIGURE 11 – Image 1973 Predicted Labels

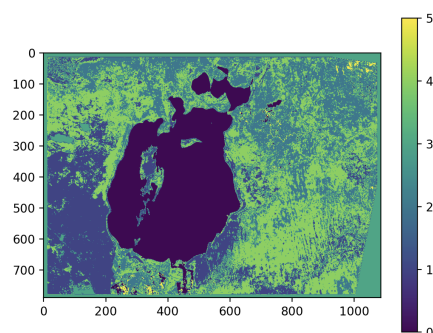


FIGURE 12 – Image 1987 Predicted Labels

On constate que plus on augmente notre nombre de cluster, moins le pourcentage du niveau de la mer est important. Il faut correctement choisir le nombre de cluster pour pas que la mer se retrouve divisé. L'avantage de 3 cluster est qu'il détecte bien la mer comme on peut le voir sur les différents graphique. Un chute de 6% du niveau de la mer entre 1973 et 1987 semble cohérent.

6 Approche non-supervisée par GMM

Question 1 : Description apprentissage du classifieur non supervisé gaussien

L'apprentissage d'un classifieur non supervisé basé sur les mélanges de gaussiennes se déroule en plusieurs étapes.

1. Initialisation des paramètres
2. Expectation-Maximization (Calculer la probabilité d'appartenance à chaque composant pour chaque données et mettre à jour les paramètres du modèle en maximisant la log-vraisemblance du jeu de données)
3. Clustering (Utiliser les responsabilités pour attribuer chaque point de données à la composante la plus probable)

Question 2 : Classifieur basé des mélanges gaussiennes

Visualisation des descripteurs d'apprentissage et de leur classe

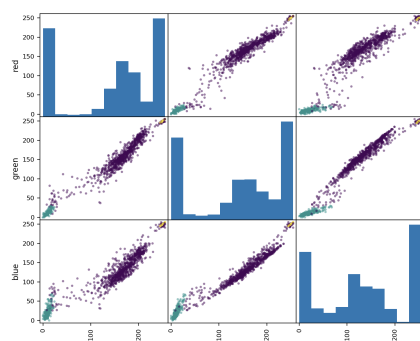


FIGURE 13 – Image 2D Gauss

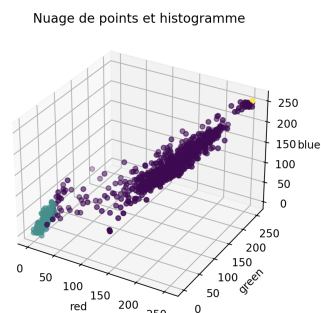


FIGURE 14 – Image 3D Gauss

Visualisation de la classification des deux images

Le code est le même, on a simplement modifié la méthode utilisée dans `unsupervisedTraining`.

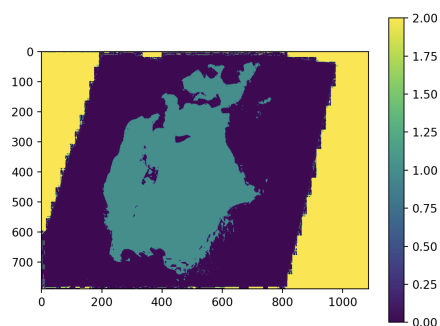


FIGURE 15 – Image 1973 Predicted Labels Gauss

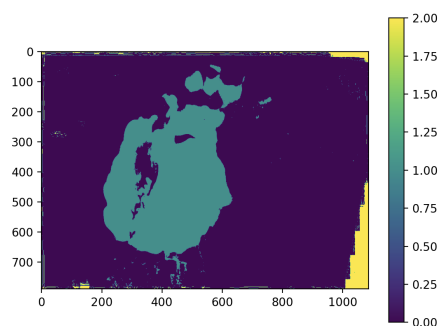


FIGURE 16 – Image 1987 Predicted Labels Gauss