# Modified Fuzzy C-Means for Remote Sensing Images Segmentation

Iacovazzi Antonio Raffaele
University of Bari Aldo Moro
a.iacovazzi6@studenti.uniba.it

September, 2022

## Abstract

The aim of this work is to understand if a modified Fuzzy C-Means algorithm (MFCM) that include also spatial information in its computation, is capable of performing good segmentation on remote sensing images. Spatial information are integrated into Fuzzy C-Means algorithm using Markov Random Field (MRF). The experiments on two subsets of different datasets show that, despite a better clusterization due to the integration of MRF, this kind of algorithm can't achieve a good accuracy when working with big aerial images since MFCM works taking into account information about the colors of pixels that often doesn't reflect properly the semantic of certain kinds of terrains.

## 1 Introduction

One of most significant task in the geospatial domain is to segment remote sensing images into different clusters which represent the kind of terrain between several classes (eg. farmlands, buildings, water, cities etc). In order to accomplish this task several methods exist, but unsupervised clustering has the big advantage that doesn't require any kind of training so it's more flexible and adaptable to different kind of territories. Fuzzy C-Mean [1] is one of most well know unsupervised clustering method that allow to assign to the same element different classes with different likelihood; in geospatial domain this feature is useful since every pixel of a remote sensed image could belong to different kind of terrain, so it's preferable to avoid crisp clustering method as K-Means in order to preserve this information. However, FCM doesn't take into account any information about the position of a pixel when works with images since it use only the information about colors. This bring the algorithm to obtain a noisy clustering, mostly when the number of cluster increase. This work is inspired by [2] where Markov Random Field are used in order to obtain more compact clusters and less noisy results. Furthermore also the kind of color space has an impact in the clustering results as shown in [3], so even this variable must be taken into account when working on a good clustering algorithm.

## 2 Related Works

Remote sensing images segmentation is a big field where a lot of different methods have been proposed both on supervised and unsupervised side.

On the supervised side there are works like [4] where ResNet50 and U-Net are used together in order to classify and segment the input, or the recent[5] where an Attention U-Net model is used to detect deforestation from aerial images obtaining important results. However, besides U-Net, other kind of models are subject of study for image segmentation task like recurrent neural network[6] and generative adversarial network[7], methodologies that could be used also on aerial images clustering.

On the unsupervised side the main challenge is to understand what kind of preprocessing is necessary to better clusterize images and how overcome the limitation of well know clustering algorithm as Fuzzy

C-Means. Several kind of modified FCM has been proposed like [8] that include spatial information in FCM or [9] that use a convolutional approach in order to accomplish the same task, or even [10] that use non local information in order to compute similarity between pixels.

# 3 Proposed Method

The Modified Fuzzy C-Means proposed in this work has been implemented using the following pipeline:

- Color space conversion (if necessary).
- Manually selection of K, the number of clusters.
- Centroid initialization based on the K-Means++ algorithm.
- Classic Fuzzy C-Means that return a fuzzy matrix $U$ and a set of centroids $V$.
- Until convergence is not reached:
  - Modified Fuzzy C-Means: calculate the probability matrix $P$ then update $U$ and $V$.
- Return the updated $U$ and $V$.
- Evaluate clustering taking for each pixel the most relevant class and comparing them to the ground truth.

Each relevant step is now described for further details.

## 3.1 Color space conversion

Colors are the various visible wavelengths that are recognizable by human eyes. Wavelength is continuous spectrum that must be discretized in order to make it processable by a computer software. Obviously there are many way to approach this task and the kind of representation used can impact in a serious manner the efficiency and the accuracy of a machine learning algorithm. The type of representation used is called "color-space" and several kind of them are available in order to catch and represent different properties of the light. In this work two types of color-space are used: the $RGB$ and the $LAB$, this in order to discover if changing the type of color-space can affect the results.

$RGB$ is the most commonly used representation where every pixel is defined as the sum of the intensity of three different colors: red, green and blue. Each pixel is a point in a tridimensional space where each component can vary in the range [0-255].

$LAB$ is a color space designed with the aim of represent all the possible colors in the entire range of human color perception. It is still a tridimensional model where the first component $L$ represent the lightness of the color and can vary in the range [0;100], $A$ represent the color component ranging from Green to Magenta and $B$ represent color component ranging from Blue to Yellow. The range of $A$ and $B$ component is not fixed and depends on the actual implementation, however goes from negative values for Green and Blue to positive values for Magenta and Yellow. In opencv, the library used in the implementation of this work $A$ and $B$ can vary in the range [-127;+127] [11].

This color space is capable of represent colors in a way that is independent from the device, furthermore, having the information about the lightness in a separate channel can influence the clustering, for this reason also this color space has been taken into account.

## 3.2 K-Means++ Centroid Initialization

Clustering algorithms like K-Means and FCM can produce different results when different centroid are randomly selected in the initialization step. This can bring to bad result and poor clustering performance and must be managed re-running the algorithm several times in order to take the best results. A good way to avoid this time-consuming process is to select the most promising centroids which are the ones that are more distant each other. This method has been proposed the first time in [12] and works as follow:

1. Take one center $c_1$, chosen uniformly at random

from X where X is the set of points.

2. Take a new center $c_i$, choosing $x \in X$ with probability
$$\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$$
where $D(x)$ denote the shortest distance from a data point to the closest center we have already chosen.

3. Repeat Step 2. until we have taken k centers altogether.

For this algorithm the sklearn implementation has been used [13]

### 3.3 Fuzzy C-Means

Fuzzy C-Means is a non-supervised clustering algorithm which is capable of assign to each point a different degree of membership to each class. Mathematically this is obtained minimizing the following objective function:

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} (u_{ik})^m d(x_i, v_k)^2 \tag{1}$$

with $m \in (1, \infty)$.

In order to achieve this, after the centroid initialization applying the already discussed K-Means++ algorithm, the following two steps must be iteratively performed:

$$u_{ik} = \frac{1}{\sum_{l=1}^{K} \left(\frac{d(x_{i,v_k})}{d(x_{i,v_l})}\right)^{\frac{2}{m-1}}} \tag{2}$$

and

$$v_k = \frac{\sum_{i=1}^{N} (u_{ik})^m x_i}{\sum_{i=1}^{N} (u_{ik})^m} \tag{3}$$

until the following condition is satisfied:

$$\max_{1 \le k \le K} (\|v_k^t - v_k^{(t-1)}\|) < \varepsilon$$

where $v_k^{(t-1)}$ is the k-th centroid computed in the previous step.

### 3.4 Markov Random Fields

Let be $X$={$X_1$,...,$X_m$} a family of random variables defined on a set $S$ which elements are organized through a neighborhood system $N$.

$X$ is said to be a Markov Random Field on $S$ with respect of the neighborhood system $N$ if and only if the followings two condition are satisfied:

$$P(x) > 0, \forall x \in X \tag{4}$$

and

$$P(x_i | x_{S-\{i\}}) = P(x_i | x_{N_i}) \tag{5}$$

where

$S - \{i\}$ is the set difference and

$x_{N_i} = \{x_{i'} | i' \in N_i\}$ are the elements neighboring the element $i$ with $i \notin N_i$ (an element cannot be a neighbour of itself).

This means that the value of $x_i$ is influenced only by its direct neighbours.

According to the Hammersley-Clifford theorem, an MRF can be equivalently characterized by a Gibbs distribution:

$$P(x) = exp(-U(x))/Z \tag{6}$$

where $Z = \sum_{x \in X} exp(-U(x))$ is a normalizing constant called the partition function and $U(x)$ is the energy function.

The energy is the sum of clique potentials $V_c(x)$ over all possible cliques $U(x) = \sum_{c \in C} V_c(x)$.

A clique is defined as a subset of sites in S where each pair of distinct sites are neighbors. $V_c(x)$ is a custom function which value depends on the local configuration on the clique $C$.

### 3.5 Modified Fuzzy C-Means

Markov Random Field are integrated into the FCM algorithm computing, at each iteration step, a tridimensional matrix that assign to each pixel the probability to belong to a certain cluster w.t.r. of its neighborhood pixels.

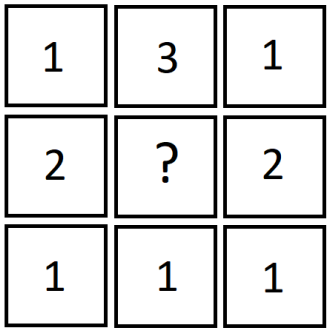
The $P$ probability matrix is computed as follows: the grid of pixels corresponds to the set $S$ and each element is labeled with a value in $X$ that represents the most likely cluster for that pixel. The possible values of $X$ are actually the cluster label.

For each pixel are computed the probabilities of each $x \in X$ using the following formula:

$$V_c(x) = \begin{cases} +1 \text{ if } x_i \neq x_{i'} \\ -1 \text{ otherwise} \end{cases} \tag{7}$$

and then applying (6) with $c \in C$ and $C$ being the set of second order clique of the pixel as shown in Figure1

Figure 1: An example of the unknown pixel label and its neighborhood (the set of second order clique).




These probabilities are then used in a modified version of (1), (2) and (3):

$$J = \sum_{i=1}^{N}\sum_{k=1}^{K}(1 - p_k(i))(u_{ik})^m d(x_i, v_k)^2 \tag{8}$$

that is the new objective function, then the two iterative formulas:

$$u_{ik} = \frac{1}{\sum_{l=1}^{K}\left(\frac{d(x_{i,v_k})(1-p_k(i))}{d(x_{i,v_l})(1-p_l(i))}\right)^{\frac{2}{m-1}}} \tag{9}$$

and

$$v_k = \frac{\sum_{i=1}^{N}(u_{ik})^m x_i(1 - p_k(i))}{\sum_{i=1}^{N}(u_{ik})^m(1 - p_k(i))} \tag{10}$$

$1 - p_k(i)$ can be considered as the resistance of neighbors $N(i)$ to assigning pixel $(i)$ the label $k$ and favor similar classes in neighboring pixels.

The MFCM algorithm needs an already computed fuzzy matrix $U$ and a set of centroids $V$, for this reason is necessary to perform the FCM before the MFCM.

## 4 Dataset Description

In order to evaluate the performance of this algorithm in remote sensing images segmentation, the following two datasets have been chosen.

The first one is the Humans in the Loop - UAE dataset [14] available on Kaggle. The dataset consists of aerial imagery of Dubai obtained by MBRSC satellites and annotated with pixel-wise semantic segmentation in 6 classes. The total volume of the dataset is 72 images grouped into 8 groups of 9 images each. Each group collect images of a certain dimension.

The classes are:

1. Building: #3C1098
2. Land (unpaved area): #8429F6
3. Road: #6EC1E4
4. Vegetation: #FEDD3A
5. Water: #E2A929
6. Unlabeled: #9B9B9B

Figure 2: Sample from the UAE dataset

In this work have been used: All the images in the first group of size (797 x 644). All the images in the fourth group of size (1.099 x 846). All the images in the eighth group of size (2.149 x 1.479). Different size groups have been selected in order to evaluate variation on performance varying the size of images.

The second one is the Gaofen Image Dataset (GID)[15], GID consists of two parts: a large-scale classification set and a fine land-cover classification set. The large-scale classification set contains 150 pixel-level annotated GF-2 images, and the fine classification set is composed of 30,000 multi-scale image patches coupled with 10 pixel-level annotated GF-2 images. GF-2 are high resolution images of size 6800 $\times$ 7200 each of them coupled with a ground truth mask.

Figure 3: Sample from the set of chunk of one image taken from the GID dataset

This dataset is thought as reference for transfer learning tasks, however, the presence of ground truth masks make it suitable also for the evaluation of unsupervised algorithms.

In this work one image from the large-scale classification set of this dataset has been used.

Given the really big size it has been splitted in 100 chunks of dimension 680 $\times$ 720, a dimension comparable to the one of the first group of UAE images. In order to make evaluation possible also the relative ground truth mask has been splitted.

The classes in the large scale classification set are:

1. Built-Up: #FF0000
2. Farmland: #00FF00
3. Forest: #00FFFF
4. Meadow: #FFFF00
5. Water: #0000FF
6. Other[1]: #000000

# 5 Experimental results

Two different experiments have been done on the two already presented dataset.The following evaluation metrics have been taken into account:

- Accuracy
- Adjusted Rand Index
- Homogeneity
- Completeness

all these metrics are computed using the ground truth as reference.

In addition to these metrics also the *Partition Coefficient* is taken into account, this is a intrinsic metric about the amount of overlap among clusters.

$$PC = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}(u_{ik})^2$$

[1]The "Other" class is not listed into the original list of classes, however is largely present into the ground truth mask so it must be reported.

The range of values for PC is $[1/k, 1]$ where $k$ is the number of clusters and a value closer to 1 means that the algorithm has well assigned each element to a certain cluster, instead a value closer to $1/k$ means that the algorithm has ended up with more fuzzier results.

The *Accuracy* measure how many points have been clustered correctly over all the clustered points. This metric is not simple to compute when used in the context of clustering because the correct match between the predicted cluster label and the ground truth label is not known in advance. For this reason a modified version of the accuracy formula is used:

$$accuracy(x, x^{'}) = \max_{perm \in P} \frac{1}{N} \sum_{i=1}^{N} 1(perm(x_i^{'}) = x_i)$$

where $1(x_i^{'} = x_i) = 1$ if $x_i^{'} = x_i$ and 0 else, $P$ is is the set of all permutations in $[1; K]$ with $K$ the number of clusters[16], so it's necessary to find the best permutation that allows to maximize the overlap between the two sets. This metric has been evaluated using the one available in the coclust library [17].

The *Rand Index*, given:

- $a$, the number of pairs of elements that are in the same cluster both in the predicted clustering and in the ground truth clustering.
- $b$, the number of pairs of elements that are in different cluster both in the predicted clustering and in the ground truth clustering.

is computed as follows:

$$RI = \frac{a + b}{\binom{N}{2}}$$

where $N$ is the total number of points. The Rand Index can tell how much the two clusterings of data have done a similar work without knows nothing a-priori about the right matching of clusters between the two sets. However the Rand Index is not always capable of properly recognize when points are clustered in a completely random way and can give high results also in these scenarios.

The *Adjusted Rand Index* is a modified version of the Rand Index which does not allow high scores comparing a totally random labelings and a ground truth labelings. Is computed discounting the expected RI $E[RI]$ of random labelings by defining the adjusted Rand index as follows:

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]}$$

and can assume values in the range $[-1, 1]$ where a negative result means that the clustering is worse than a random clustering (is somehow orthogonal), 0 means that there is no correlation and a positive value means that there is a correlation between the two clusterings. A value of 1 means a perfect score where all the elements are clustered in the same way.

The *Homogeneity* is the measure of how much each predicted cluster has points belonging to the same ground truth class.

The *Completeness* measure how much each points belonging to the same ground truth class are clustered into the same cluster.

Homogeneity and completeness scores are formally given by:

$$h = 1 - \frac{H(C|K)}{H(C)}$$

$$c = 1 - \frac{H(K|C)}{H(K)}$$

where $C$ is the set of ground truth clusters, $K$ is the set of predicted clusters, $H(X)$ is the entropy function of $X$ and $H(X|Y)$ the conditional entropy function of $X$ given $Y$ [18].

## 5.1 Algorithms hyper-parameters

The followings hyper-parameters have been chosen for both the experiments:

- Number of max iteration: 1000, both from FCM and MFCM. After this iteration the algorithm stops since it could have fallen into a local minima.
- m (fuzzification coefficient): 2, both from FCM and MFCM.
- epsilon: 0.01 for FCM and 0.2 for MFCM.

## 5.2 First Experiment

The first experiment was done on the previously described subsets of the Human in the Loops dataset[14] called the *small* subset, the *medium* subset and the *big* subset. $K = 6$ has been chosen since it is the most probable number of classes in each ground truth mask and every image has been processed both in RGB and LAB color space obtaining different results. All the metrics have been computed for each image followed by a macro averaging of the metrics, grouped by image size.

The following tables collect the results for each different subset.

| Algorithm | Accuracy | ARI | Homog. | Complet. |
|---|---|---|---|---|
| FCM - RGB | 0.4358 | 0.1976 | 0.3840 | 0.2341 |
| FCM - LAB | 0.4375 | 0.2054 | 0.3912 | 0.2417 |
| MFCM - RGB | 0.4776 | 0.2267 | 0.4058 | 0.2543 |
| MFCM - LAB | 0.4586 | 0.2243 | 0.4115 | 0.2581 |

Table 1: Macro-averaged results for the small subset

| Algorithm | Accuracy | ARI | Homog. | Complet. |
|---|---|---|---|---|
| FCM - RGB | 0.3339 | 0.0495 | 0.1312 | 0.1007 |
| FCM - LAB | 0.3368 | 0.0541 | 0.1317 | 0.1008 |
| MFCM - RGB | 0.3593 | 0.0631 | 0.1569 | 0.1210 |
| MFCM - LAB | 0.3658 | 0.0697 | 0.1586 | 0.1219 |

Table 2: Macro-averaged results for the medium subset

| Algorithm | Accuracy | ARI | Homog. | Complet. |
|---|---|---|---|---|
| FCM - RGB | 0.3864 | 0.1466 | 0.2503 | 0.1804 |
| FCM - LAB | 0.3980 | 0.1527 | 0.2487 | 0.1795 |
| MFCM - RGB | 0.4209 | 0.1760 | 0.2775 | 0.2000 |
| MFCM - LAB | 0.4357 | 0.1865 | 0.2769 | 0.2002 |

Table 3: Macro-averaged results for the big subset

| Algorithm | Partition Coeff. | Exec. Time |
|---|---|---|
| FCM - RGB | 0.7033 | ~15 min |
| FCM - LAB | 0.7013 | ~11 min |
| MFCM - RGB | 0.9740 | ~15 min |
| MFCM - LAB | 0.9750 | ~11 min |

Table 4: Intrinsic evaluation results for the small subset

| Algorithm | Partition Coeff. | Exec. Time |
|---|---|---|
| FCM - RGB | 0.6815 | ~15 min |
| FCM - LAB | 0.6765 | ~12 min |
| MFCM - RGB | 0.9448 | ~15 min |
| MFCM - LAB | 0.9434 | ~12 min |

Table 5: Intrinsic evaluation results for the medium subset

| Algorithm | Partition Coeff. | Exec. Time |
|---|---|---|
| FCM - RGB | 0.7070 | ~160 min |
| FCM - LAB | 0.7027 | ~105 min |
| MFCM - RGB | 0.9601 | ~160 min |
| MFCM - LAB | 0.9594 | ~105 min |

Table 6: Intrinsic evaluation results for the big subset

Figure 4: MFCM - LAB clustering for the same sample of the UAE Figure2

Looking at these results it's possible to notice that the Modified Fuzzy C-Means performs better in every situation and the LAB color space often improves

result a bit and mostly reduce computation time[2]. However the overall performance in comparison with the ground truth are poor, probably because the color-based clustering is not capable of catch similarity in the texture of terrains that can vary its colours and still belong to the same category; this happen often in the reference dataset. On other hands the partition coefficient is good for the MFCM, so the algorithm is capable to well distinguish the elements in the images. The execution time increase drastically when working with big images, this also because the algorithm has fallen quite often in the local minima trap dealing with that subset.

### 5.3 Second Experiment

The second experiment was done on the 100 chuncks of one image taken from the previously described subsets of the GID[15] dataset. $K = 3$ has been chosen since it is the average number of clusters present into the ground truth masks and every image has been processed both in RGB and LAB color space obtaining different results. All the metrics have been computed for each images followed by a macro averaging over all 100 results.

| Algorithm | Accuracy | ARI | Homog. | Complet. |
|---|---|---|---|---|
| FCM - RGB | 0.6046 | 0.1539 | 0.1019 | 0.0811 |
| FCM - LAB | 0.5860 | 0.1489 | 0.1042 | 0.0776 |
| MFCM - RGB | 0.6454 | 0.1553 | 0.0943 | 0.0922 |
| MFCM - LAB | 0.6156 | 0.1624 | 0.1047 | 0.0885 |

Table 7: Macro-averaged results for all the 100 chunks.

[2]Computation time has the same value for FCM and MFCM because has been computed when both the process stops for each image. However FCM times is the most relevant, during the execution it was the one which influence mostly the computation time.

| Algorithm | Partition Coeff. | Exec. Time |
|---|---|---|
| FCM - RGB | 0.8303 | ˜250 min |
| FCM - LAB | 0.8157 | ˜105 min |
| MFCM - RGB | 0.9881 | ˜250 min |
| MFCM - LAB | 0.9857 | ˜105 min |

Table 8: Intrinsic evaluation results for the 100 chunks processing.

Figure 5: MFCM - RGB clustering for the same sample of the GID chuncked Figure3

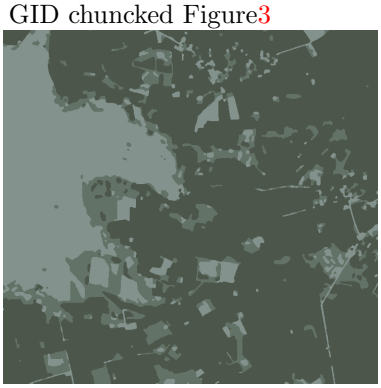
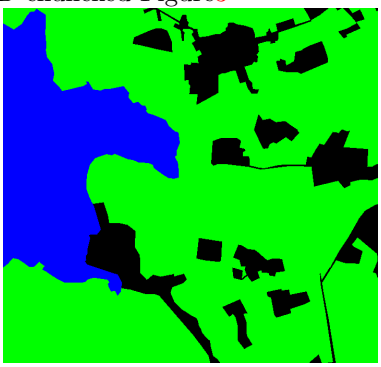


Figure 6: Ground Truth mask for the same sample of the GID chuncked Figure3



Looking at these results it's possible to notice that the accuracy increased substantially but the adjusted Rand index, the homogeneity and the completeness are lower than before, this probably because having

less clusters was possible to find a good permutation between classes but the actual correlation between the predicted cluster and the ground truth is somehow missed. In the example it's possible to see how the water is well recognized, but farmlands and "other" classes are mixed. This probably for the same reason of the first experiment, the two classes are clustered by texture similarities that are not recognized using a color based clustering. Furthermore for this experiment the LAB color space does not help to improve results, rather, it contribute to lower them, however it still improves the execution time.

# 6 Conclusions and future works

The experiments have shown that for the remote sensing images segmentation this algorithm has a lot of space for improvement but also that the inclusion of spatial relationship into the clustering algorithm improve to keep similar pixel into the same cluster. Results by [2] and [9] has shown an overall accuracy over the 70% but proper comparison with these works are not possible without the same images and their ground truth masks. As already said one major fault of a color based clustering algorithm is that is not capable of take into account the textures, or patterns that could give the same semantics even if their color are different, as in the farmland example of the GID dataset.

Possible improvement on this work could be implement a way to easily detect the proper number of clusters, try to expand the windows of neighbour in the Markov random field in order to understand if can further increase performance and trying to work using other color space such as HSV. Further studies can be done on a different evaluation metrics of the distance between point in a way that can include also pattern information but its probably a task that require semi-supervised or supervised techniques.

# References


[1] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," 1983. 1

[2] Y. HongLei, P. JunHuan, X. BaiRu, and Z. DingXuan, "Remote sensing classification using fuzzy c-means clustering with spatial constraints based on markov random field," 2013. 1, 9

[3] H. Tariq and S. Burney, "K-means cluster analysis for image segmentation," 2014. 1

[4] P. Ulmas and I. Liiv, "Segmentation of satellite imagery using u-net models for land cover classification," 2020. 1

[5] D. John and C. Zhang, "An attention-based u-net for detecting deforestation within satellite sensor imagery," 2022. 1

[6] F. Visin, A. Romeroy, M. Ciccone, K. Kastnery, K. Choz, M. Matteucci, Y. Bengioy, and A. Courvilley, "Reseg: A recurrent neural network-based model for semantic segmentation," 2016. 1

[7] P. L. andC amille Couprie and S. Chintala, "Semantic segmentation using adversarial networks," 2016. 1

[8] K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen, "Fuzzy c-means clustering with spatial information for image segmentation," 2006. 2

[9] A. Singh, A. Kumar, and P. Upadhyay, "Modified possibilistic c- means with constraints (mpcm-s) approach for incorporating the local information in a remote sensing image classification." 2020. 2, 9

[10] F. Zhao, L. Jiao, and H. Liu, "Fuzzy c-means clustering with non local spatial information for noisy image segmentation," 2011. 2

[11] opencv. Color conversions. [Online]. Available: https://docs.opencv.org/3.2.0/de/d25/imgproc_color_conversions.html 2

[12] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," 2007. 2

[13] sklearn. K-means++ sklearn implementation. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.kmeans_plusplus.html 3

[14] H. in The Loop. Satellite images of dubai, the uae. [Online]. Available: https://www.kaggle.com/datasets/humansintheloop/semantic-segmentation-of-aerial-imagery 4, 7

[15] X.-Y. Tong, G.-S. Xia, Q. Lu, H. Shen, S. Li, S. You, and L. Zhang, "Land-cover classification with high-resolution remote sensing images using transferable deep models," 2022. [Online]. Available: https://x-ytong.github.io/project/GID.html 5, 8

[16] S. Morbieu. Accuracy: from classification to clustering evaluation. [Online]. Available: https://smorbieu.gitlab.io/accuracy-from-classification-to-clustering-evaluation/#accuracy-for-clustering 6

[17] F. Role, S. Morbieu, and M. Nadif. Coclust: a python package for co-clustering. [Online]. Available: https://coclust.readthedocs.io/en/v0.2.1/ 6

[18] sklearn. Clustering performance evaluation. [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation 6