



Cartographie des changements d'Utilisation et occupation des Terres pour la stratification

Version février 2021

Historique de versions	
novembre 2020	Version initiale
Février 2021	1.1 : mise à jour du script pour la séparation des échantillons de validation et d'entraînement
Mars 2021	1.2 <ul style="list-style-type: none">- Choix des images, mise a jour du script earthengine- Addition de détails dans la classification sur la détermination des classes d'UOT

Objectifs : cet SOP détaille les procédures pour la création d'une carte d'utilisation et d'occupation des terres et de ces changements en vue de préparer un échantillonnage probabiliste aléatoire stratifié

Introduction :

Puisque l'utilisation de la carte serait de créer des zones homogènes pour le placement des points d'échantillonnage lors de l'évaluation des émissions et des réductions des émissions, la carte serait optimisée pour l'élimination des erreurs d'omissions sur le changement de la forêt (dégradation, régénération). On serait donc amené à mettre en évidence et détecter les strates de changements au mieux que possible.

Procédures

Les différentes étapes suivantes devraient alors être suivies pour la consistance des données, mais surtout pour minimiser les différences qui pourraient être causées par le changement des opérateurs dans les futures implémentations de l'exercice.

Les étapes dans la création de la carte sont :

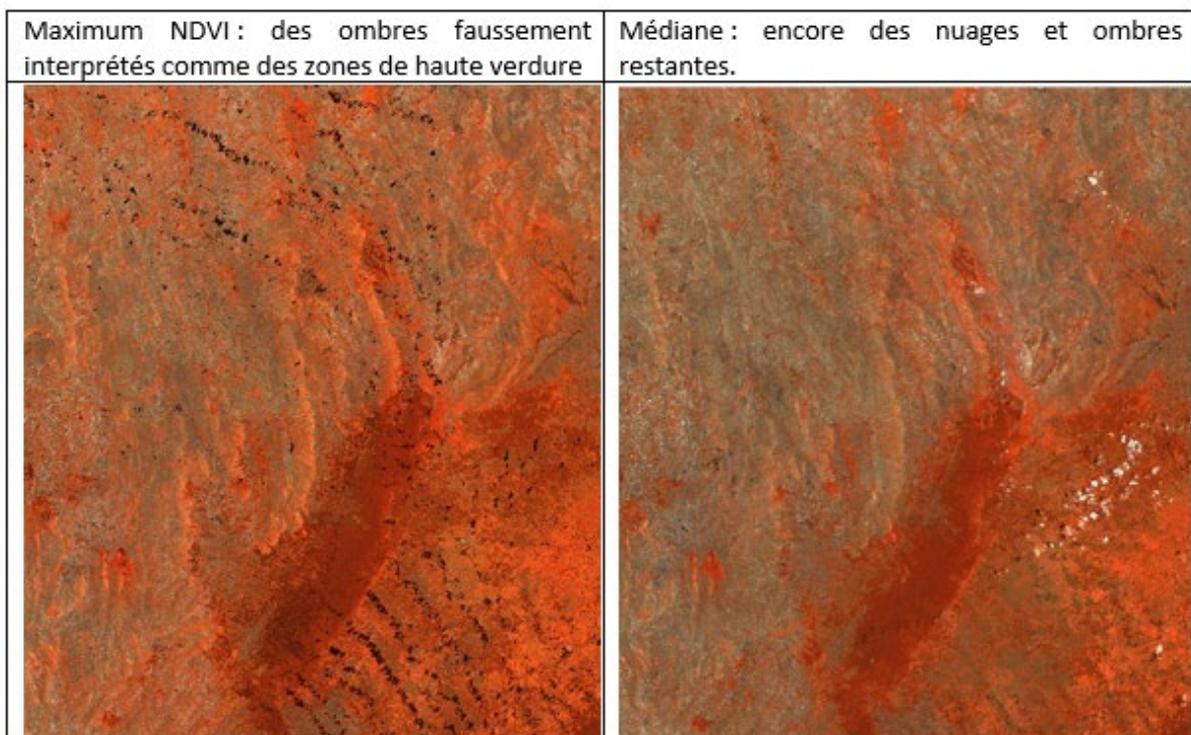
- Choix des images

- Choix des classes d'utilisation et occupation des terres et du changement
- Etablissement des zones d'entraînements
- Classification/segmentation
- Evaluation des erreurs / contrôle qualité

Choix des images

Les meilleures images pour la classification sont ceux en début de saison de pluie. En saison sèche, les nuages fins sont trop fréquents et presque impossibles de trouver des images de qualité, et en saison humides, les gros nuages sont aussi fréquents. Cette période correspond aussi au début de la période végétative et le contraste entre forêt et non forêt est évident. Aussi, pour tenir compte du cycle annuel des activités de feux et déforestation, on va faire une image composite de décembre en juin.

Pour l'algorithme de filtrage des nuages, on aura le choix entre utiliser la médiane de la collection d'image ou l'image au pixel le plus vert (greenest pixel).



En utilisant les images sentinel -2 de traitement LEVEL1, on obtient une collection de plus d'images, et donc un meilleur composite. Le script suivant crée la composite, il suffit de changer la date et la tuile pour obtenir chacune des dates et des tuiles.

```
//importation des variables
var s1 = ee.ImageCollection("COPERNICUS/S2");
var roi = ee.FeatureCollection("users/env_rs/roi39kta");
var tile39KTA = ee.FeatureCollection("users/env_rs/tuiles_39KTA");
var visu =
{"opacity":1,"bands":["classification"],"min":0,"max":7,"palette":["00ff00",
"00ff00","ff0000","aaaa00","aaaa00","aaaa00","0000ff"]};

//travailler sur tuile 39KTA
```

```

//changer selon la tuile

//definition des dates
var date2 = [ee.Date("2020-12-01"),ee.Date("2021-03-05")];
var date1 = [ee.Date("2019-12-01"),ee.Date("2020-06-30")];

//liste des bandes a utiliser (ou exporter)
var optical = ["B2","B3","B4","B5","B6","B7","B8","B8A","B11","B12"];

//selection des images dans 39KTA ERPAA selon la date
//et nuages <10 pourcent
var tile39tka_date1 = s1
  .filterDate(date1[0],date1[1])
  .filter(ee.Filter.eq("MGRS_TILE","39KTA"))
  // .filterBounds(erpaa)
  //.filter(ee.Filter.lt("CLOUDY_PIXEL_PERCENTAGE",30))
  ;

print("collection sentinel ",tile39tka_date1);

//calcul NDVI et ajouter comme bande
//creation fonction qui calcule NDVI et ajoute comme bande
var addNDVI = function(image) {
  var ndvi = image.normalizedDifference(['B8', 'B4'])
    .rename('NDVI')
    .multiply(10000)
    .toUint16();
  //remplacer NDVI>0.90 par 0
  //var ndvi = ndvi0.where(ndvi0.gt(0.90),0);
  return image.addBands(ndvi);
};

//appliquer la formule sur la collection d'image
var tile39tka_date1_ndvi = tile39tka_date1.map(addNDVI);

print("collection sentinel avec NDVI ",tile39tka_date1_ndvi);

//creer l'image greenest pixels
var tile39tka_date1_greenest = tile39tka_date1_ndvi.qualityMosaic("NDVI");

var tile39tka_date1_greenest = tile39tka_date1_greenest.select(optical);
var tile39tka_date1_greenest = addNDVI(tile39tka_date1_greenest);

print("greenest pixel date1 ",tile39tka_date1_greenest);

//*****
//  date2
//*****

//selection des images dans 39KTA ERPAA selon la date
//et nuages <10 pourcent
var tile39tka_date2 = s1
  .filterDate(date2[0],date2[1])
  .filter(ee.Filter.eq("MGRS_TILE","39KTA"))
  // .filterBounds(erpaa)
  //.filter(ee.Filter.lt("CLOUDY_PIXEL_PERCENTAGE",30))
  ;

print("collection sentinel date2 ",tile39tka_date2);

```

```

//appliquer la formule sur la collection d'image
var tile39tka_date2_ndvi = tile39tka_date2.map(addNDVI);

print("collection sentinel avec NDVI 2 ",tile39tka_date2_ndvi);

//creer l'image greenest pixels
var tile39tka_date2_greenest = tile39tka_date2_ndvi.qualityMosaic("NDVI");

var tile39tka_date2_greenest = tile39tka_date2_greenest.select(optical);
var tile39tka_date2_greenest = addNDVI(tile39tka_date2_greenest);

print("greenest pixel 2 ",tile39tka_date2_greenest);

//*****
// Classification
//*****

//compiler les images (stack)
var inImage = tile39tka_date1_greenest.addBands(tile39tka_date2_greenest);
print("image multiband",inImage);

//training
var training = inImage.sampleRegions({
  collection: roi,
  properties: ['Id'] ,
  scale:10
});
//print(training, 'training');

//creation de l'objet randomForest
// Making a Random Forest classifier and training it.
var rf = ee.Classifier.smileRandomForest(200).train({
  features: training,
  classProperty: 'Id',
  //inputProperties: bands
});

print("le model randomForest",rf);

//classification de l'image
var uot39tka = inImage.classify(rf);
print("image classifie", uot39tka);

//exporter l'image dans drive
Export.image.toDrive({
  image: tile39tka_date1_greenest,
  description: "tuile39KTA_date2",
  scale: 10,
  region: tile39KTA,
  maxPixels:2e9});

Export.image.toDrive({
  image: uot39tka,
  description: "stratif",
  scale: 10,
  region: tile39KTA,
  maxPixels:2e9});

Map.centerObject(tile39KTA,10);

```

```
//Map.addLayer(tile39tka_date1_greenest, {min:10,max:3000,bands: ["B6", "B5", "B4"]}, "date1");
//Map.addLayer(tile39tka_date2_greenest, {min:10,max:3000,bands: ["B6", "B5", "B4"]}, "date2");
Map.addLayer(quot39tka, visu, "stratification");
```

Choix des classes d'utilisation du sol et de changement

Le plus important pour l'analyse des émissions serait la cartographie des conversions. On devrait donc bien capturer les classes avec conversions. On aurait les classes suivantes :

- Forêt persistante (FF)
- Déforestation (FN)
- Régénération de forêt (NF)
- Non forêt restant non-forêt (NN)
- Plan d'eau restant plan d'eau (WW)

Pour information, la catégorie N (non forêt) ici regroupe les catégories C (Zones cultivées), G (savanes), O (zones nues), et S (Surfaces artificielles).

Etablissement des zones d'entraînements

Pour mieux gérer les itérations de classification, on va créer différents shapefiles pour chacun des classes d'OUT. Faire des polygones sur des endroits homogène et représentatifs. Commencer par les classes de persistances car elles seront plus faciles à trouver. Les cartes d'UOT existant peuvent être utilisés pour faciliter cette étape :

- Carte UOT produite par LOFM BNCCR pour ERPAA
- Carte de déforestation par LOFM/BNCCR (disponible prochainement)
- Carte Déforestation 2000-2018 par USFS
- Carte de changement de couverture forestière par Rakotomalala et al.
- Carte de la déforestation (webmap) de WRI/GFW

Après la recherche des classes persistantes, on essaye de faire les classes de changements, en général près des zones anciennement reforestées, et des bordures de forêt. La même chose pour la régénération s'il y a lieu. Pour faciliter aussi le choix des zones de changement, soustraire la NDVI des deux images, là où c'est positif, on a des gains, et là où c'est négatif, on a des pertes.

Le formule pour la NDVI pour les images Sentinel 2 est

$$NDVI := \text{Index}(B8, B4) = \frac{B8 - B4}{B8 + B4}.$$

Il faut nommer les différents shapefiles des sites d'entraînement aux code UOT plus haut, on aura donc **FF.shp**, **FN.shp**, **NF.shp**, **NN.shp** et **WW.shp**, et les mettre dans le répertoire **input**

Classe forêt persistante

La forêt persistante est facile à identifier, avec la couleur grise rougeâtre sur l'image multi date et rouge foncée avec la fausse couleur (sur les deux dates). On crée des zones homogènes pour les zones

d'entraînement, donc il serait nécessaire d'avoir plusieurs classes de forêt persistante pour bien délimiter et avoir un bon résultat pour la forêt.

Non forêt persistante

Toutes les parties qui ne sont pas identifiées comme forêt seront des non-forêt. Pourtant, on devrait créer plusieurs zones d'entraînement homogène et donc plusieurs classes de non-forêt pour que la classification soit propre et n'empiètera pas sur la forêt. Il faut s'assurer que les classes sont homogènes, même si on devrait créer plusieurs classes

Eau

L'eau est facile à détecter visuellement, mais il y a seulement si peu de surface d'eau qu'il est parfois nécessaire de parcourir toute l'image pour avoir assez de zones d'entraînement. Aussi, les zones inondables avec beaucoup de végétation (rizières, bord des grands lacs) pourraient être confondues avec la forêt, de même pour les parties avec ombres assez importantes. Prendre en compte ces diversités et créer les zones d'entraînement en rapport avec cela. Deux ou trois classes pour l'eau serait peut-être suffisantes.

Déforestation

Pour faciliter l'identification de la déforestation et des pertes en forêts, on mettra sur la vue (en QGIS ou ArcGIS l'image de date 1, l'image de date 2 et l'image combinée multi-date. Les images uniques sont affichées en fausse couleur Infrarouge. Plus c'est rouge et foncé, plus dense est la végétation.

Ensuite, l'image multi date est affiché de façon à mettre en évidence les changements dans la bande proche infrarouge. En faisant une composition colorée de RGB (bande8.1-Bande8.1,Bande8.2), on voit en rouge les pertes de biomasses, et donc la déforestation. Les zones sans changement apparaîtront en gris.



Figure 1 : image fausse couleur infrarouge

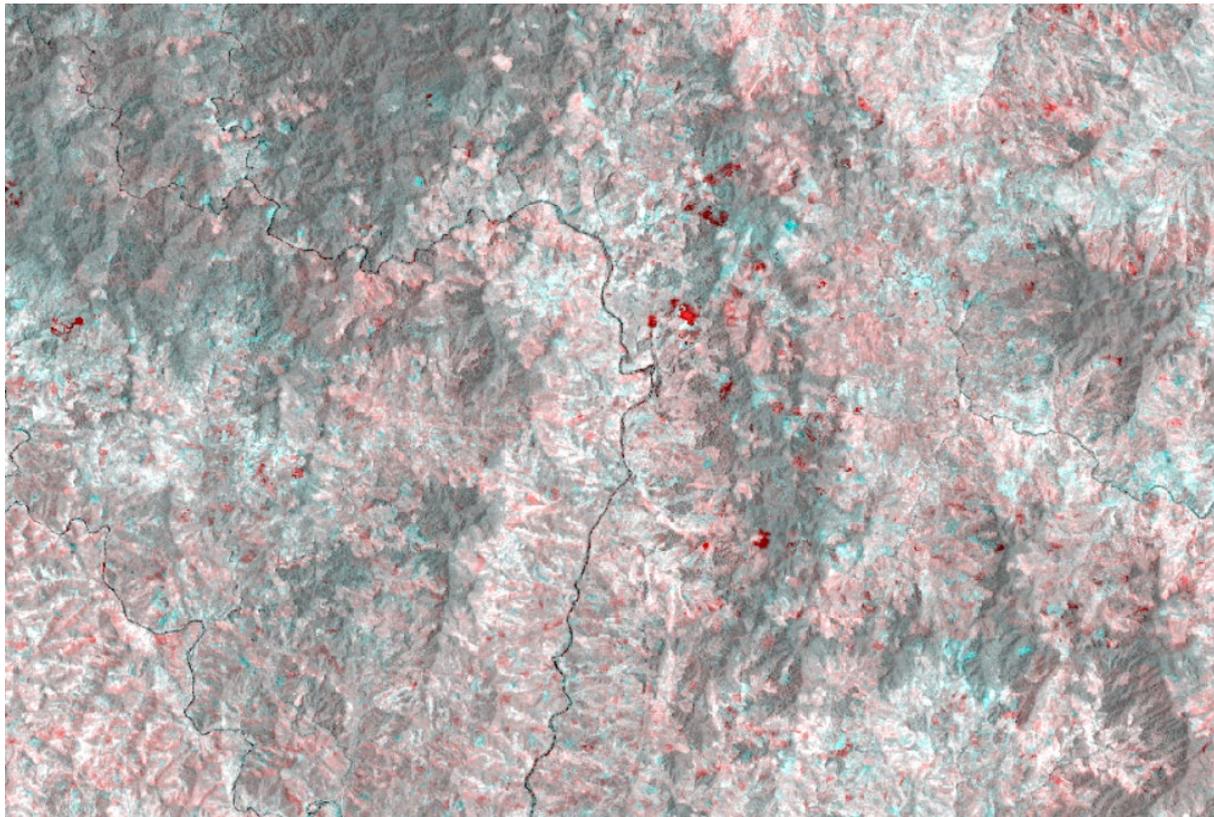


Figure 2 : Image de changement en Proche infrarouge sur les 3 bandes RGB

Classification

Pendant cette étape, on devrait donc déjà avoir :

- Les images pour la date de référence et la date d'évaluation
- Les shapefiles contenant les zones d'entraînement, un shapefile par classe

Créer un répertoire classification, et mettre le script dedans, puis créer un sous-répertoire input et output. Les données en entrées pour le script seront dans le répertoire input, et les résultats dans le répertoire output. La structure du répertoire devrait maintenant ressembler à la Figure 1

› A12023 › Documents › classification

Name	Date modified	Type	Size
input	01/12/2020 14:54	File folder	
output	01/12/2020 14:54	File folder	
classification.R	20/10/2020 14:21	R File	3 KB

Figure 3 : Structure de dossier

Prendre le script et faire les changements nécessaires pour les noms des fichiers en entrée et les noms de fichiers de sortie. Installer les package suivantes si ce n'est pas déjà fait :

- Raster
- Rgdal

- Xlsx
- spatialEco

copier et exécuter le script suivant, il faut changer les noms des entités en entrée et les noms de fichier de sortie

```

# importer les modules

library(randomForest)
library(raster)
library(rgdal)
library(xlsx)
library(spatialEco)

setwd("F:/sops/sop0")

#lire shapefile
#readOGR(lien_vers_le_fichier,nom_du_fichiers_ans_extension)

roi_forest <- readOGR(dsn = "./input",layer="FF")
roi_defor <- readOGR(dsn = "./input",layer="FN")
roi_nonfor <- readOGR(dsn = "./input",layer="NN")
roi_water <- readOGR(dsn= "./input",layer="WW")

#verifier les fichiers importes
#plot(roi_water)

roi_all <- bind(roi_forest,roi_defor,roi_nonfor,roi_water)

#plot(roi_all)

#lire une image raster
s2_image2 <- brick("input/part1_raw_2020.tif")
s2_image1 <- brick("input/part1_raw_2015.tif")

#plot(s2_image1)
s2_bandnames <- c('B1', 'B2', 'B3',
'B4', 'B5', 'B7', 'ndvi1', 'ndvi2', 'ndvi3', 'srtm')
names(s2_image1) <- s2_bandnames
names(s2_image2) <- s2_bandnames

#combiner les deux dates en une seule image
inImage <- stack(s2_image1,s2_image2)

#point dans les polygones
set.seed(16)
sample_ff <- spsample(roi_forest,2500,"random")

to_train <- sample(1:2500,2000)
to_valid <- (1:2500)[-to_train]

training_ff <- sample_ff[to_train]
valid_ff <- sample_ff[to_valid]

sample_fn <- spsample(roi_defor,2500,"random")
training_fn <- sample_fn[to_train]
valid_fn <- sample_fn[to_valid]

```

```

#sample_nf <- spsample(roi_gain,2000,"random")
sample_nn <- spsample(roi_nonfor,2500,"random")
training_nn <- sample_nn[to_train]
valid_nn <- sample_nn[to_valid]

sample_wv <- spsample(roi_water,2500,"random")
training_wv <- sample_wv[to_train]
valid_wv <- sample_wv[to_valid]

#on peut combiner les points maintenant
point_sample <- bind(training_ff,training_fn,training_nn,training_wv)

#re-obtenir les attributs
point_sample <- point.in.poly(point_sample,roi_all)

#attaching raster attributes to shapefile points
training_point <- extract(inImage,point_sample)

# creation du modele randomforest
rf <-
randomForest(x=training_point,y=as.factor(point_sample@data$classuot),ntree
= 201,importance = TRUE)

#####
# Creation de la carte de changement, changez le nom
#####

predict(inImage, rf, filename="output/fcc_date1date2.img",
type="response",index=1, na.rm=TRUE, progress="window", overwrite=TRUE)

#####
#validation #####
#####

point_valid <- bind(valid_ff,valid_fn,valid_nn,valid_wv)

#ajouter les attributs
point_valid <- point.in.poly(point_valid,roi_all)#

#re-importer l'image classifiee
classified <- brick("output/fcc_date1date2.img")

#attaching raster attributes to shapefile points
validation <- extract(classified,point_valid)
point_valid <- data.frame(point_valid,validation)

write.csv(table(point_valid$classuot,point_valid$Layer_1),"output/confusion
_matrix.csv")

confusion_matrix <- table(point_valid$classuot,point_valid$Layer_1)

#omission error

ff_omission <-
(confusion_matrix[2,1]+confusion_matrix[3,1]+confusion_matrix[4,1])/500*100

```

```

fn_omission <-
(confusion_matrix[1,2]+confusion_matrix[3,2]+confusion_matrix[4,2])/500*100
nn_omission <-
(confusion_matrix[1,3]+confusion_matrix[2,3]+confusion_matrix[4,3])/500*100
ww_omission <-
(confusion_matrix[1,4]+confusion_matrix[2,4]+confusion_matrix[3,4])/500*100

omission_error <-c(ff_omission,fn_omission,nn_omission,ww_omission)

#comission error

ff_comission <-
(confusion_matrix[1,2]+confusion_matrix[1,3]+confusion_matrix[1,4])/500*100
fn_comission <-
(confusion_matrix[2,1]+confusion_matrix[2,3]+confusion_matrix[2,4])/500*100
nn_comission <-
(confusion_matrix[3,1]+confusion_matrix[3,2]+confusion_matrix[3,4])/500*100
ww_comission <-
(confusion_matrix[4,1]+confusion_matrix[4,2]+confusion_matrix[4,3])/500*100

comission_error <-c(ff_comission,fn_comission,nn_comission,ww_comission)

erreur <- data.frame(c('ff','fn','nn','ww'),omission_error,comission_error)
names(erreur) <- c('class','omission','comission')

#ecrire les resultats en excel
write.xlsx(erreur,"./output/erreur.xlsx")
write.xlsx(confusion_matrix,"./output/matrice_confusion.xlsx")

```

il suffit maintenant de récupérer le résultat dans le répertoire output et voir si on n'a pas fait de grossière erreur. Si c'est le cas, il faut revoir les sites d'entraînement et refaire le processus.

Post traitement :

Une fois la classification est satisfaisante, on passe à la correction manuelle des imperfections restantes sur l'image. On va comparer la classification avec la carte d'UOT de ERPAA. On effectuera les modifications suivantes :

- Si on a des déforestations dans la carte de stratification, alors qu'on a des non -forêt sur ERPAA, alors, on changera en non-foret
- Si on a des forêts sur la carte de stratification alors qu'on a des non-forêts dans l'UOT, on changera en non-forêt
- Toutes les parties Eaux dans la carte ERPAA sera rapportée en tant que Eaux sur la carte de stratification

Ce changement serait effectué par un script R, qui demande en entrée la carte UOT ERPAA, déjà reclassifiée et la carte de classification. Le produit serait une carte de la déforestation, corrigée des erreurs comme les déforestations identifiées sur les zones déjà non-forêts, ou les ombres reportées comme eaux.

```

##
##Ce script elimine les deforestation
##qui se trouve sur les zones deja non foret
##utiliser apres classification final
##la carte d'utilisation du sol PREAA est la base

```

```

##transformee en forêt, non foret, eau
##le resultat est foret, deforestation, non foret eau

#####
#demande en entrée :
# - raster bibande, avec bande 1 = erpaa recodee, bande 2 = la
classification
# - code 11 = foret, 12 = deforesatation, 22 = non foret, 44 = eaux
# le script prendra la deforestation dans la classification, et appliquer
# a l'image erpaa (utilisee comme base)
# les deforestation daans les zones deja non foret sont juste changes en
non foret

#importer les libraries
library(raster)
library(rgdal)

#parametres initiales
#definir les couleurs de la palette pour affichage
mycolor <- c("#80C0DB", "#00FF00", "#FF0000", "#F0E6E6", "#0000FF")
#nom de fichier de sortie (modifier si necessaire) il sera dans le rep
output
fichier <- "output/post_tk39.tif"

#repertoire de travail
setwd("F:/sops/sop0")

#stacker les 2 raster d'abord en utilisant QGIS ou ArcGIS
#pour avoir deux raster de meme taille et de meme projection
#importer l'image a deux bandes

img_suivi <- stack ("39kta_erpaa_classif.tif")

#renommer les bandes de l'image
names(img_suivi) <- c("erpaa", "classif")

#fonction conditionnel pour utiliser avec overlay
condit <- function(erpaa, classi){
  0*(erpaa==0)+ #les zones erpaa = 0 resterons 0
  11*(erpaa == 11 & classi!=2)+ # les zones foret de ERPAA resteront foret
sauf
  12*(erpaa == 11 & classi == 2)+ # les zones deforestation de classif
  22*(erpaa == 22)+ # les zones non foret de erpaa
  44*(erpaa == 44) # les zones eaux de erpaa, resterons eaux
}

result <- overlay(img_suivi, fun=(condit))

plot(result, breaks=c(0,1,11,12,22,44), col=mycolor)

writeRaster(result, fichier, format="GTiff", overwrite=TRUE, datatype ="INT2U")

```

On passera aussi un filtre (sieve) de 49 pixels pour éliminer les artéfacts de classification, et qui éliminera les entités de moins de 0.5 hectares. Pendant le filtrage, les pixels de déforestation ne seront pas filtrés.

Evaluation de qualité

La qualité de la classification est évaluée utilisant la matrice de confusion, et en calculant les erreurs d'omission et des erreurs de commission. Ces informations sont enregistrées par le script dans le repertoire **output**

Drive (E:) > sops > sop0 > output

Name	Date modified	Type	Size
 fcc_date1date2.img	04/12/2020 19:11	Disc Image File	387,029 KB
 fcc_date1date2.img.aux.xml	04/12/2020 19:11	XML Document	1 KB
 training_point.dbf	04/12/2020 17:48	DBF File	1,934 KB
 training_point.prj	04/12/2020 17:48	PRJ File	1 KB
 training_point.shp	04/12/2020 17:48	SHP File	165 KB
 training_point.shx	04/12/2020 17:48	SHX File	47 KB
 fcc_date1date2.rrd	04/12/2020 19:11	RRD File	129,943 KB
 erreur.xlsx	04/12/2020 22:22	Microsoft Excel W...	4 KB
 matrice_confusion.xlsx	04/12/2020 22:23	Microsoft Excel W...	4 KB

Figure 4 : structure du repertoire output

Dans le fichier Excel erreur se trouve les informations calculées sur les erreurs tandis que la matrice de confusion est le tableau originel utilisé pour calculer les erreurs. Ce qui est important à noter est la valeur d'omission et de commission pour la classe de changement. Ces chiffres devraient être suffisamment petit pour qu'on puisse utiliser la carte.

Tableau 1 : Erreur d'omission et de commission

Classe	Omission (%)	Commission (%)
ff	4.4	4.6
fn	7	1.4
nn	1.4	7
ww	0.2	0

Dans notre cas, on a donc sous-évalué la déforestation de 7% et surévalué de 1.4% en même temps. Une carte contiendra toujours des erreurs, et qu'on décide si on pourrait maintenant utiliser la carte ou non compte tenu des erreurs énumérés dans le Tableau 1. On va essayer surtout de ramener l'erreur d'omission de la déforestation le plus proche de 0 possible, quitte à avoir une erreur de commission de déforestation augmenter un peu. Le seuil acceptable pour omission de déforestation serait de 0.2%. Si on est satisfait, on passe à l'étape de la stratification.