

Défense d'APP

Apprentissage machine

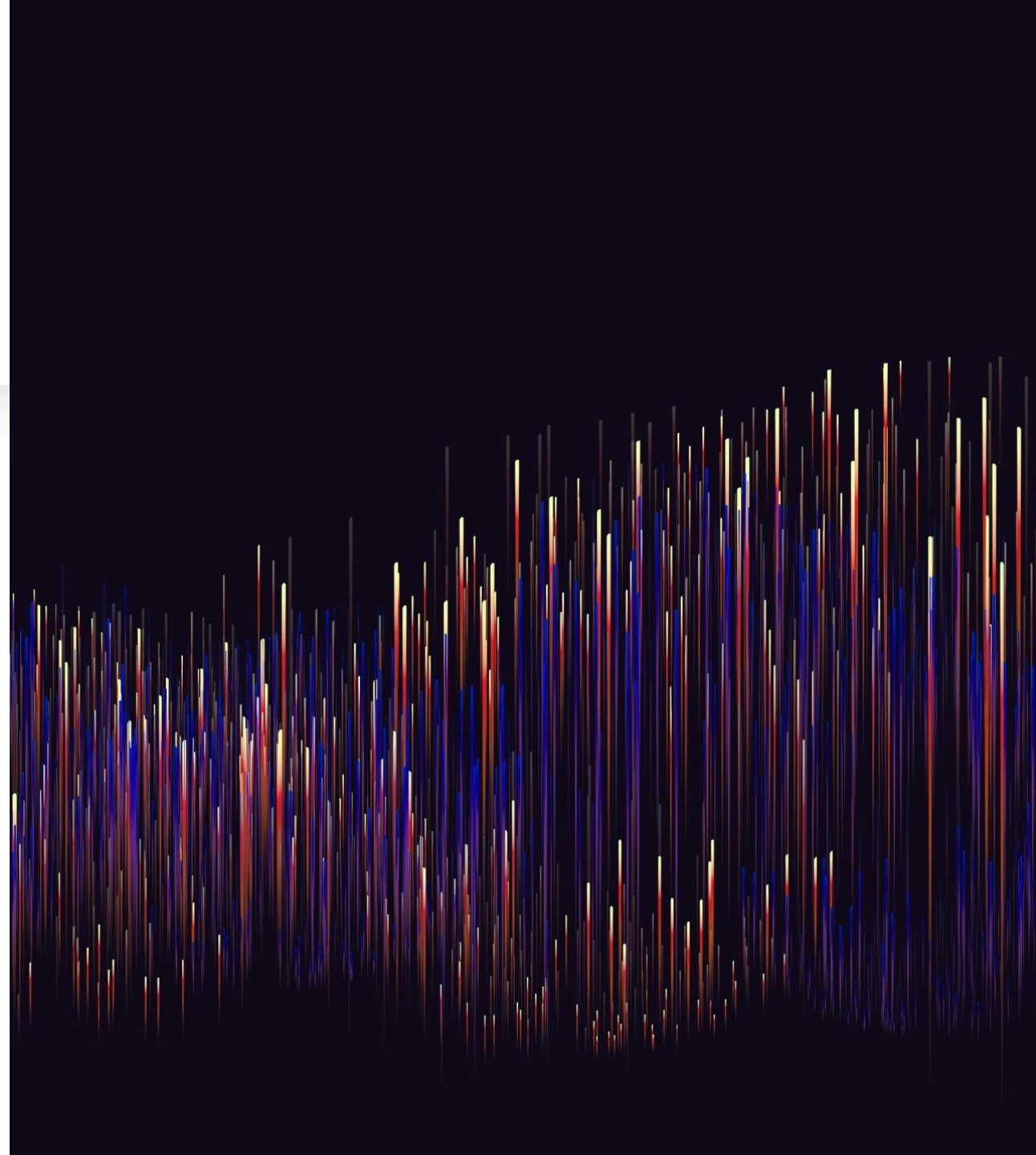
Jacty Milena Saenz Rosales – saej3101

Laurence Milette – mill3003

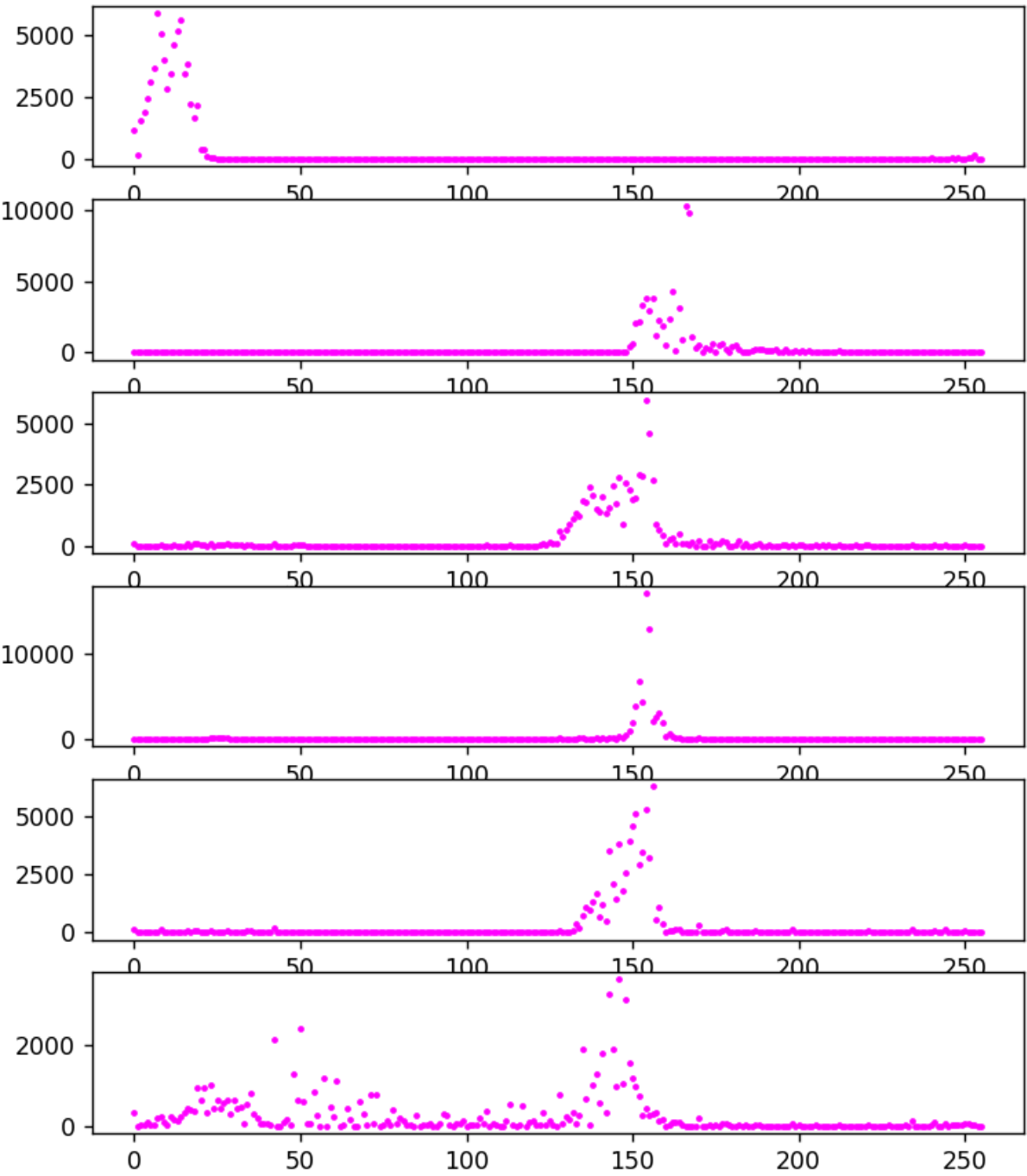
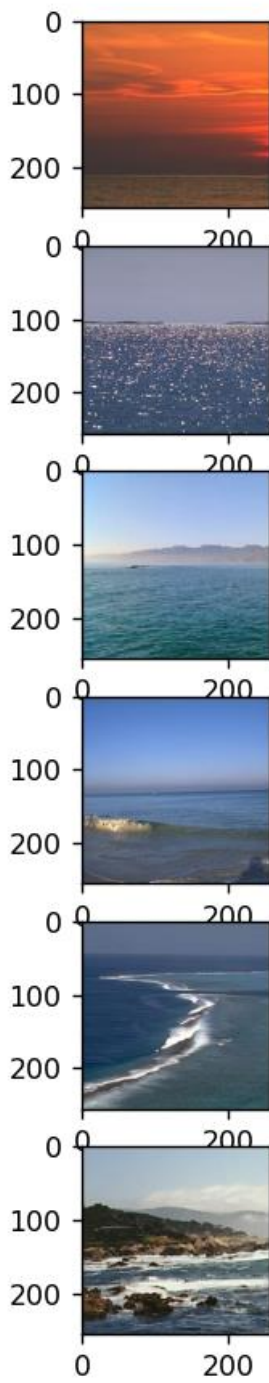
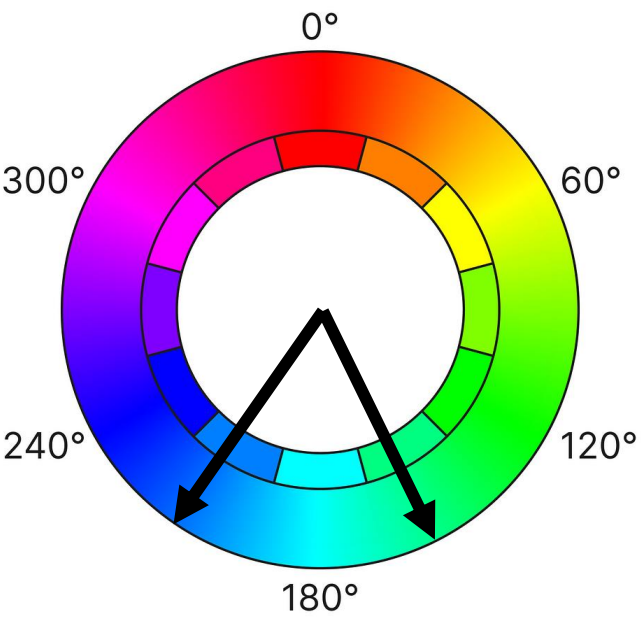
Anthony Royer – roya2019

Représentation

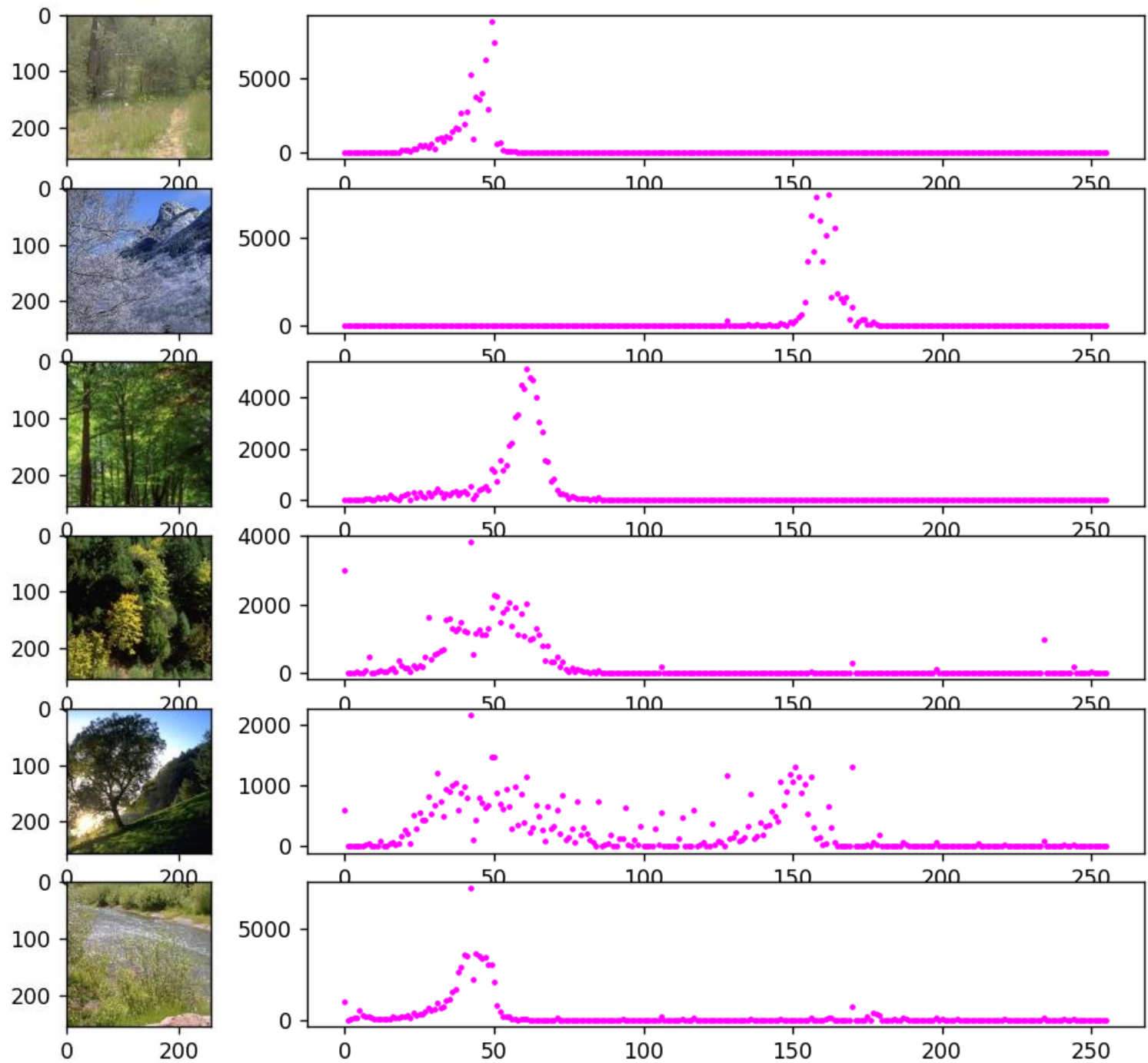
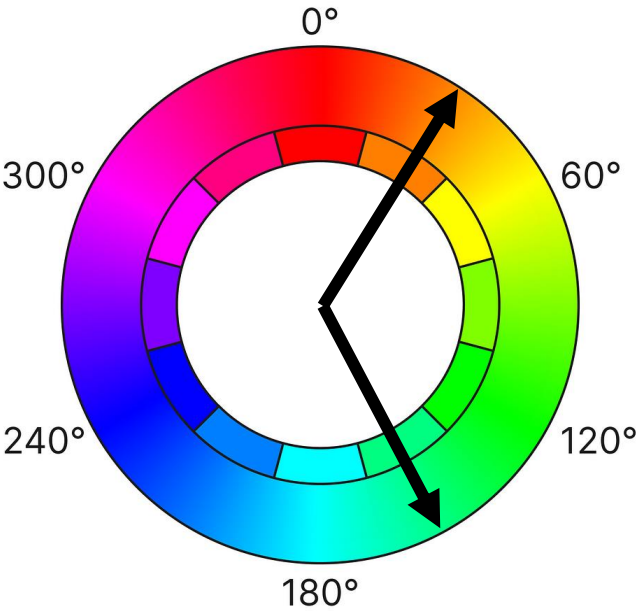
- Liste de plusieurs caractéristiques tel que:
 - RGB
 - HSV
 - Teinte (Hue)
 - Saturation
 - Détection de contours (Edge detection)
 - Détection de lignes
- 3 techniques choisis
 - **Teinte (Hue)**
 - **Lignes verticales**
 - **Autres lignes** (autres que verticales et horizontales)



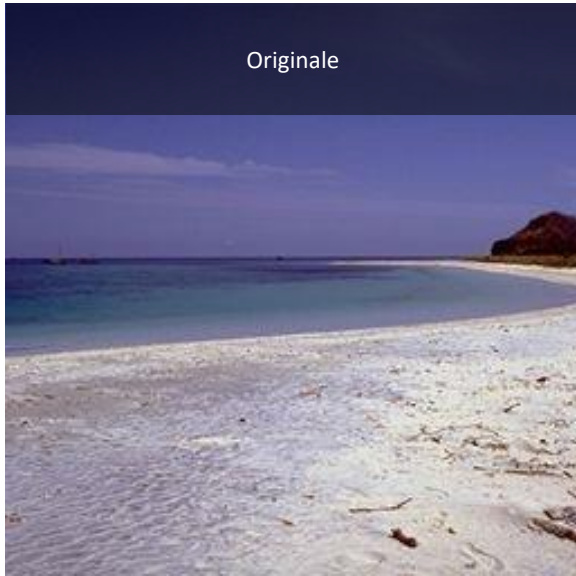
Teinte - Plages



Teinte - Forêts



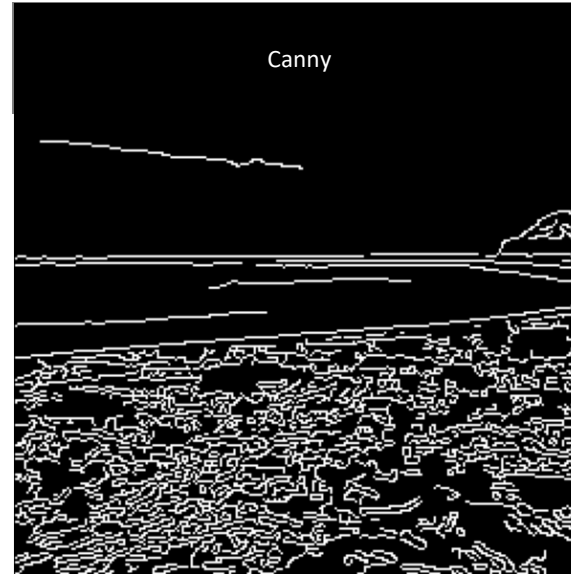
Originale



Gris



Canny



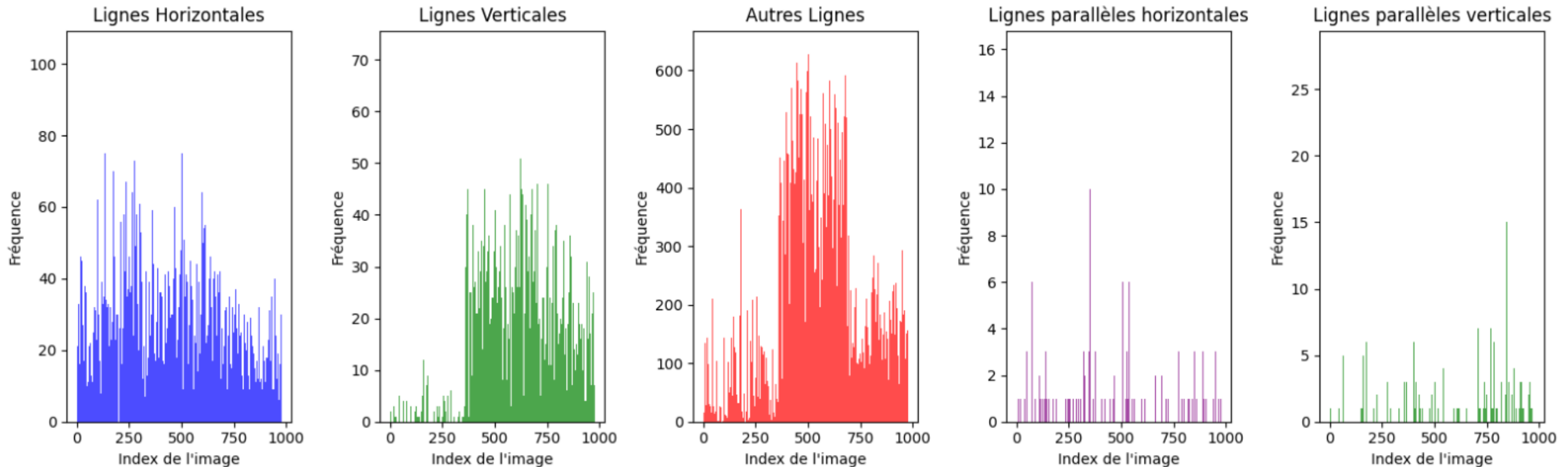
Lignes avec Hough



Processus de détection de lignes

Processus de détection de lignes

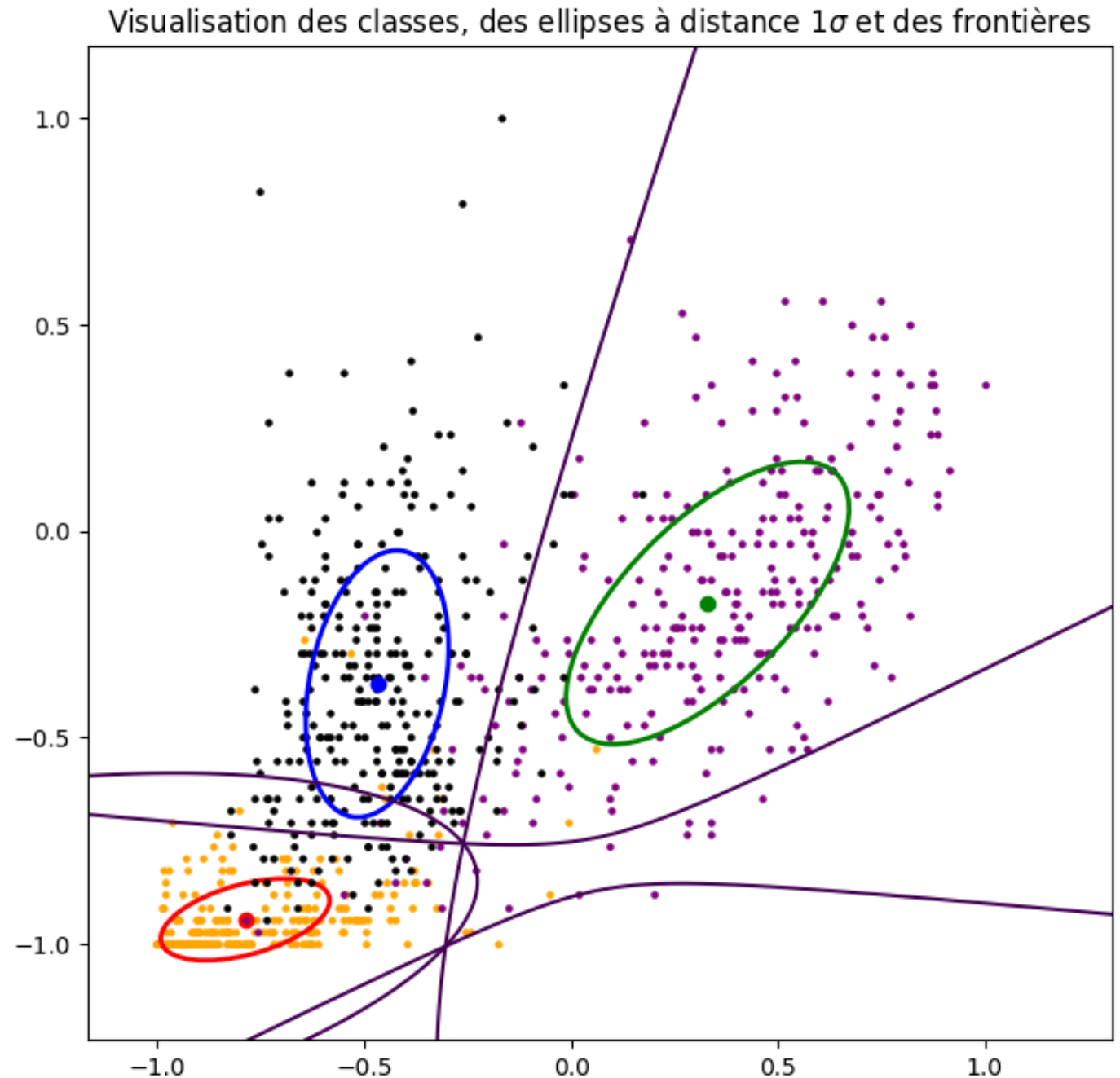
Histogramme des types de lignes dans les images

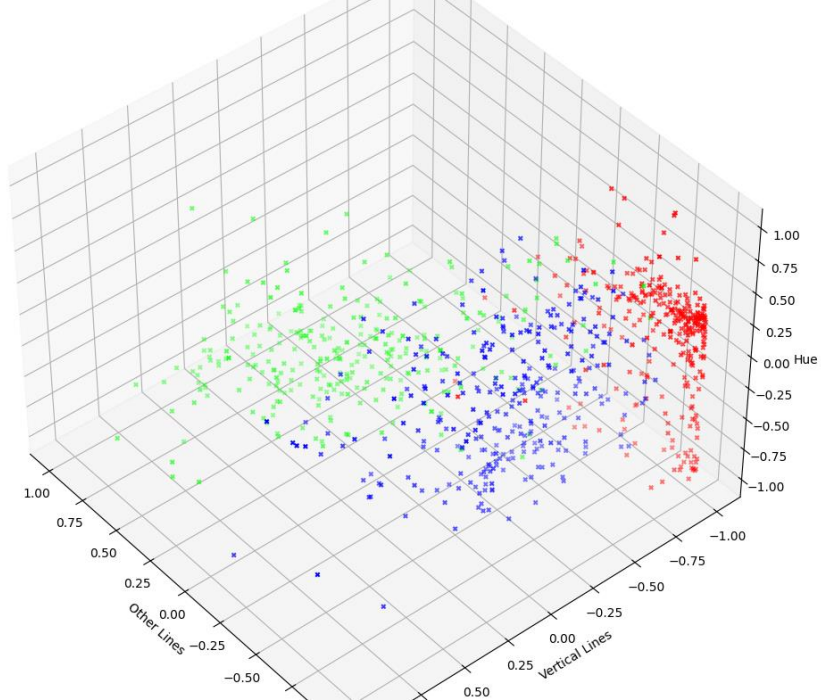


Réprésentation

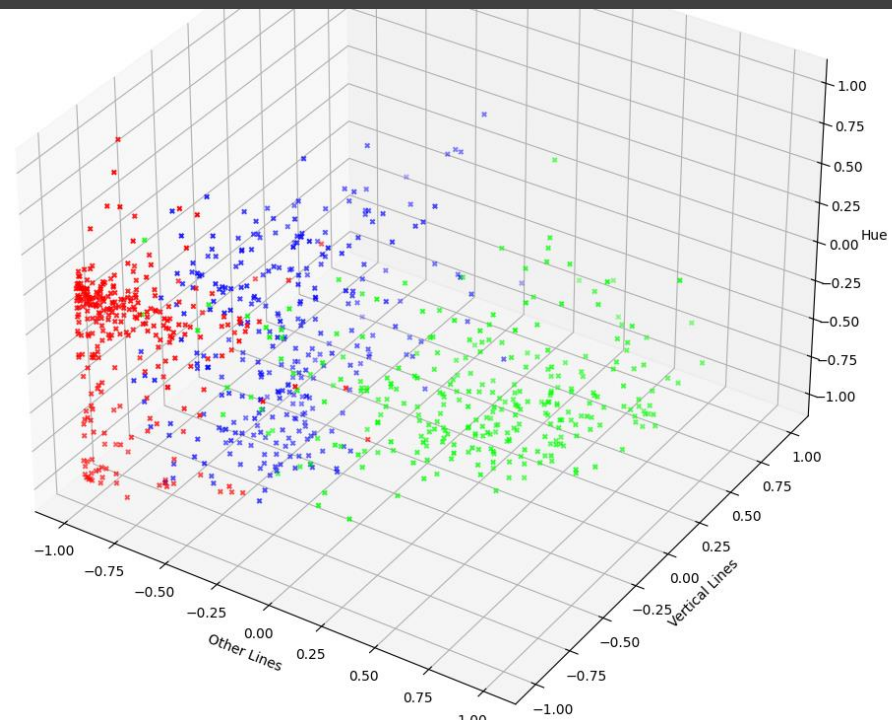
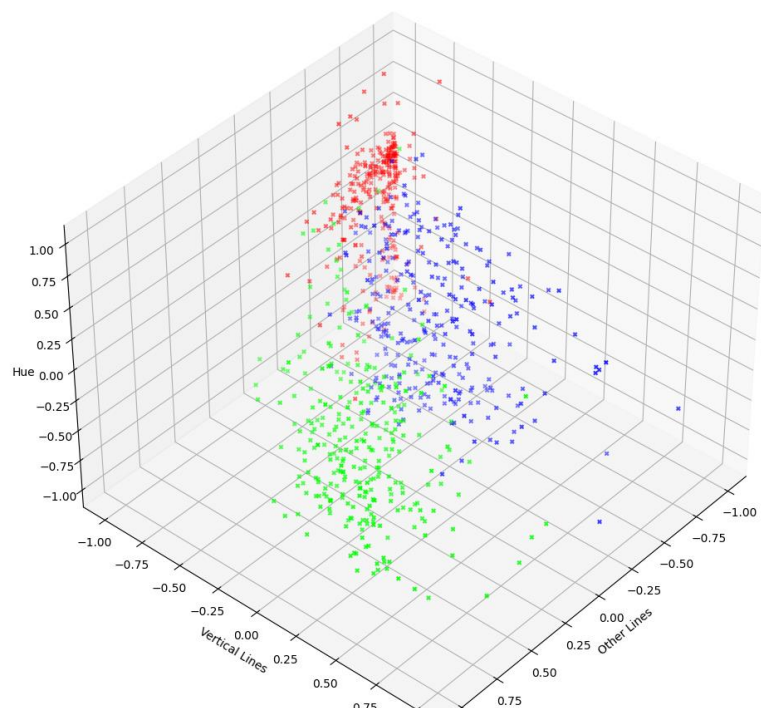
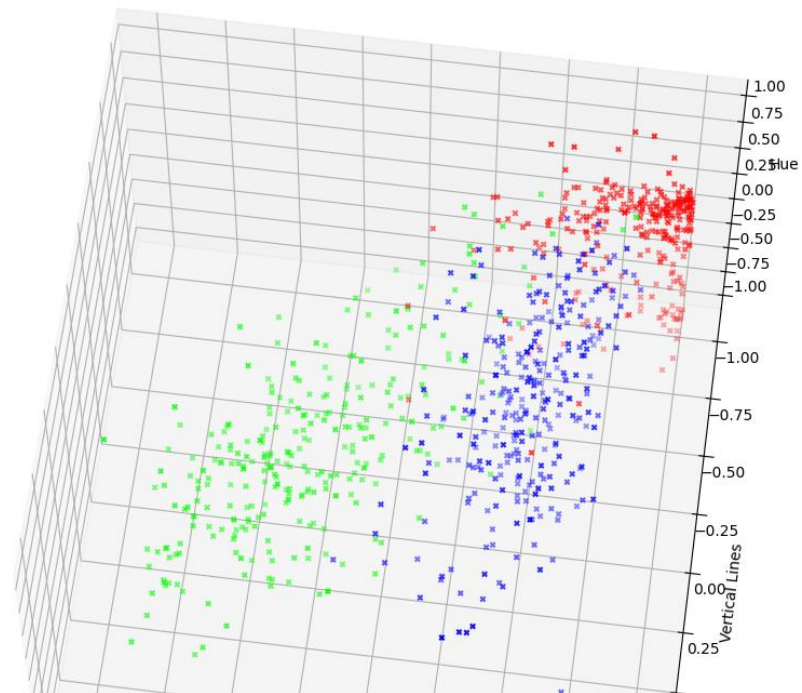
- Moyenne des valeurs maximales du Hue
- Lignes verticales
- Autres lignes

- Plages
- Forêts
- Villes



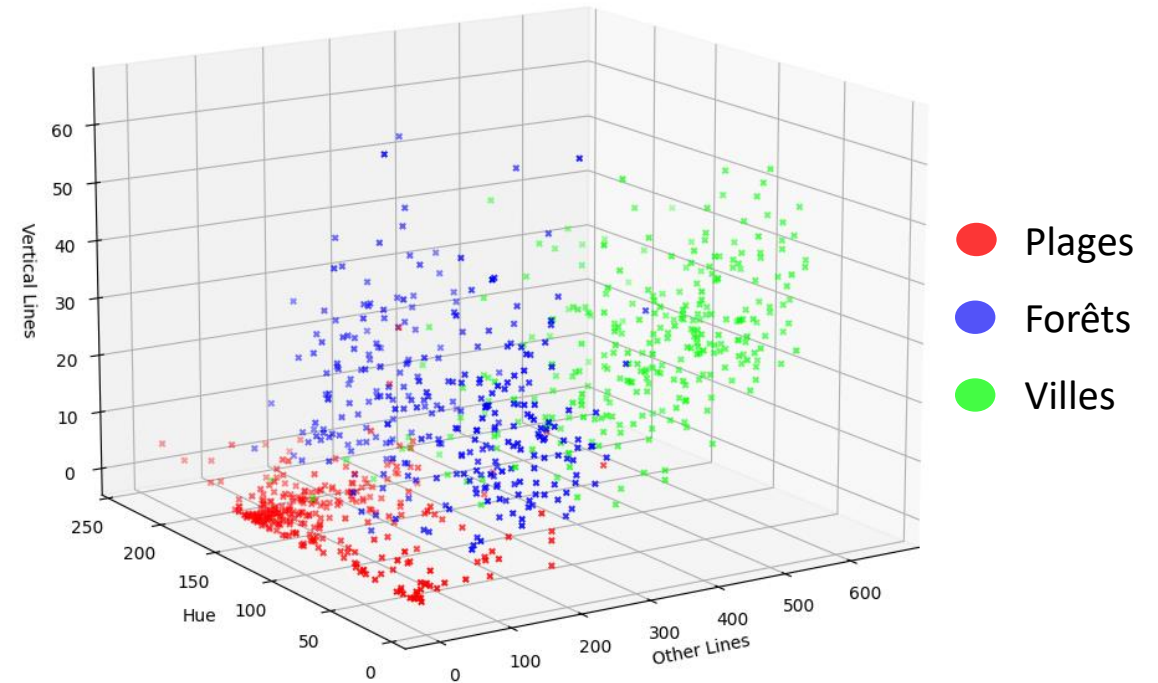


- Plages
- Forêts
- Villes



Prétraitement

- Ensemble de données
- Normalisation
- Essaies décorrélation (non idéale)



Classificateur Bayésien

Probabilité Gaussienne

Choix des hyperparamètres

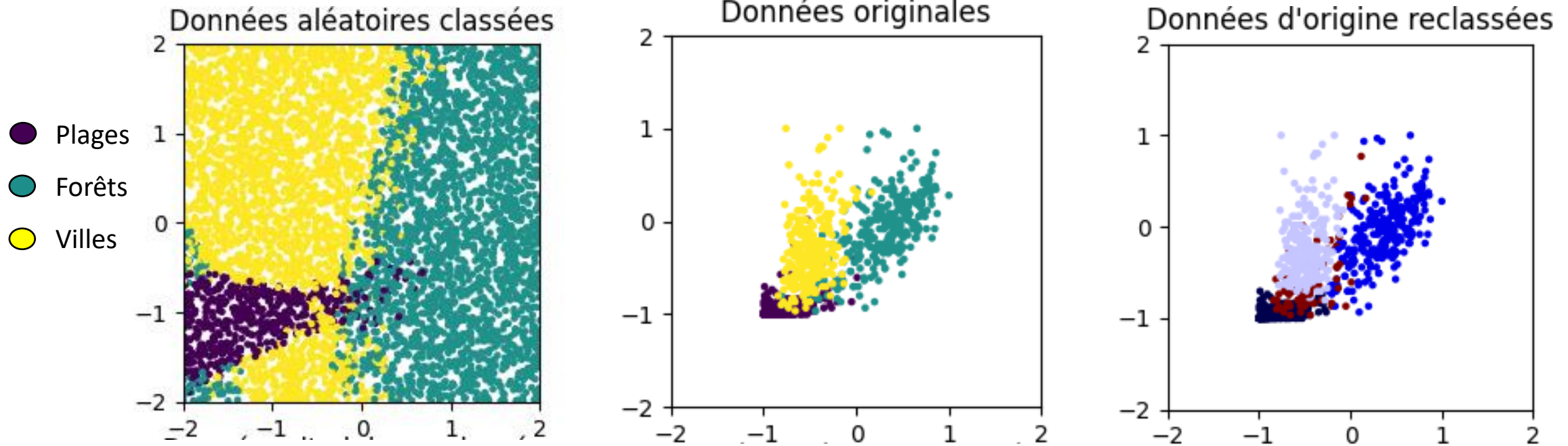
Hyperparamètres	Valeurs
Modèle	gaussien
Matrice de coûts	$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix}$

Matrice de confusion

$$\begin{bmatrix} 272 & 4 & 14 \\ 8 & 266 & 16 \\ 23 & 7 & 260 \end{bmatrix}$$

Classificateur Bayésien

Probabilité Gaussienne



Classificateur k-PPV

Choix des hyperparamètres

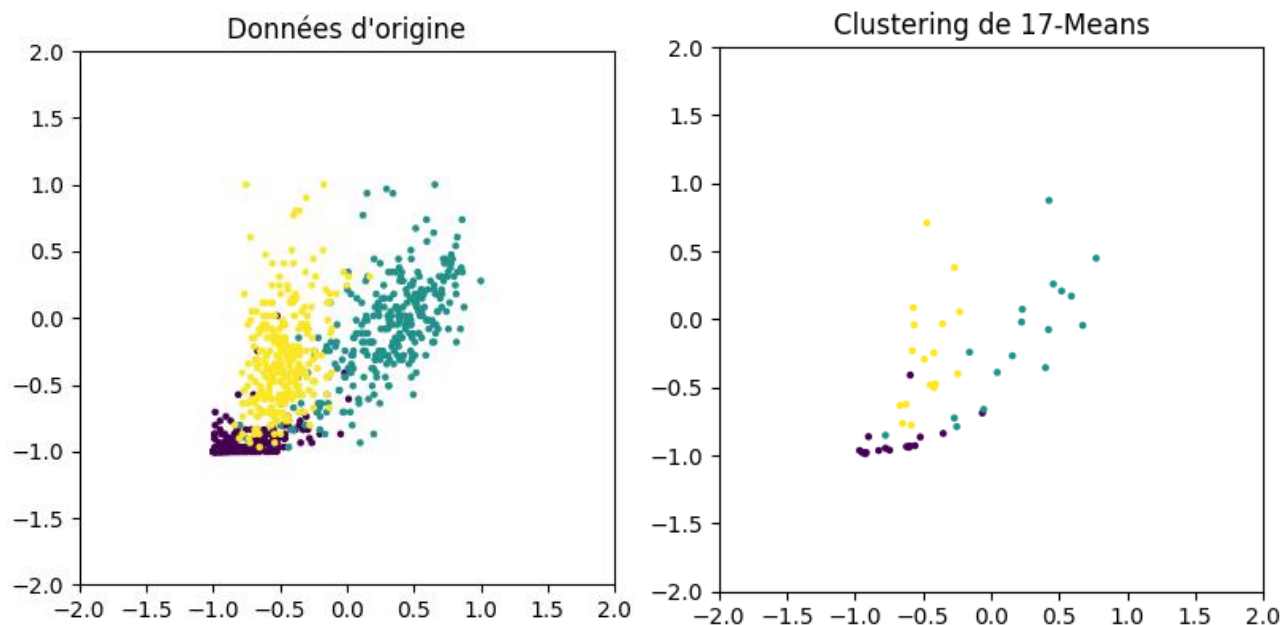
Hyperparamètres	Valeurs
Nombre de voisins	5
Nombre de représentants	17

Matrice de confusion

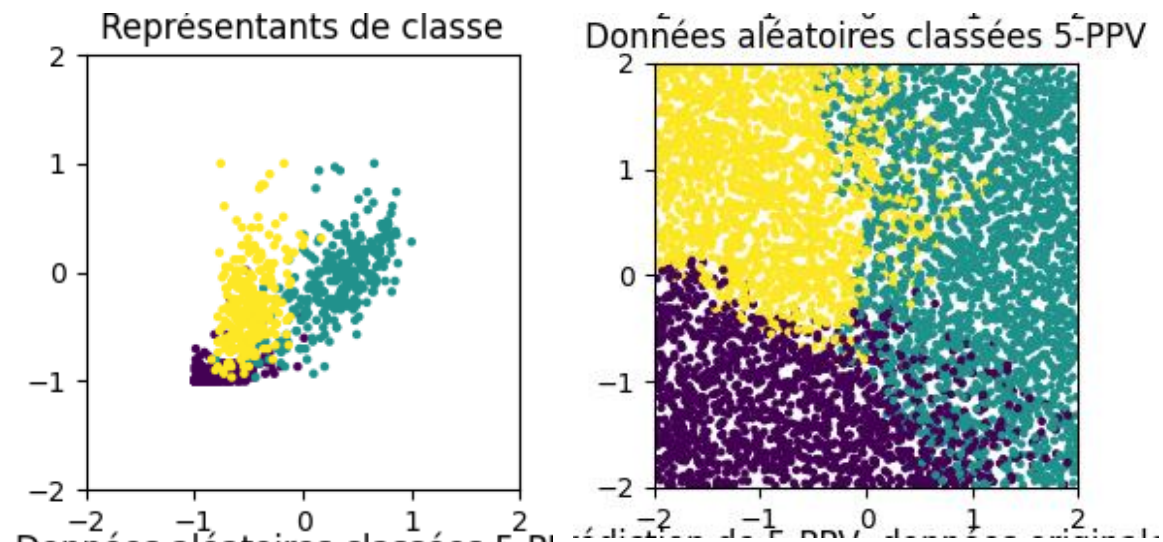
$$\begin{bmatrix} 284 & 4 & 2 \\ 11 & 266 & 13 \\ 53 & 10 & 227 \end{bmatrix}$$

Classificateur k-PPV

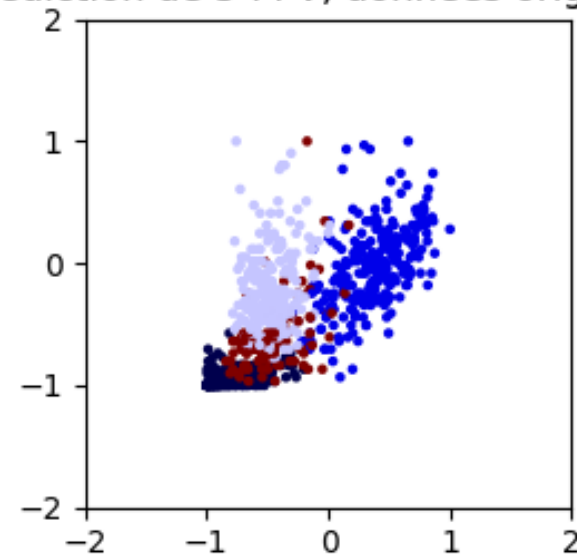
Représentants pour 5-PPV sur le 17-moy



- Plages
- Forêts
- Villes



Prédiction de 5-PPV, données originales



Classificateur Réseaux Neurones Artificielle (RNA)

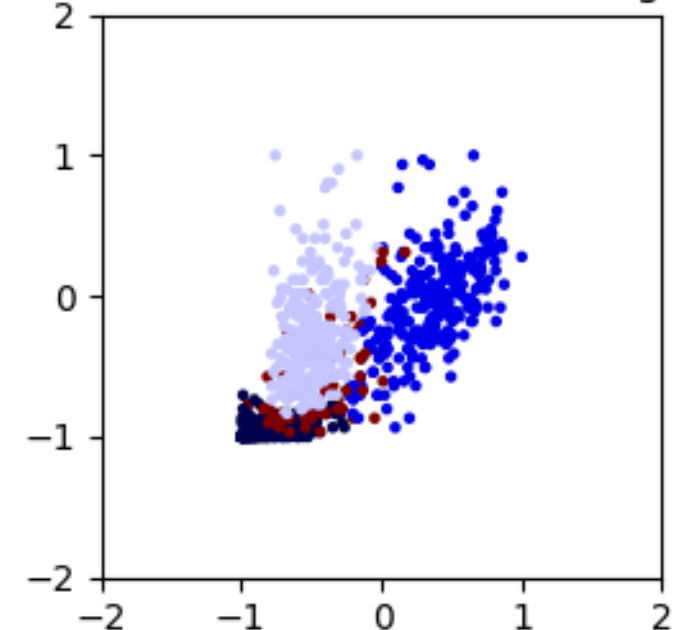
Choix des hyperparamètres

Hyperparamètres	Valeurs
Fonction d'activation	<i>Tanh</i>
Nombre de couches	9
Nombre de neurones	10
Nombre d'épochs	1500
Taux d'apprentissage	0,0002
Calcul du <i>loss</i>	<i>categorical</i>

Matrice de confusion

$$\begin{bmatrix} 267 & 7 & 16 \\ 4 & 269 & 17 \\ 11 & 9 & 270 \end{bmatrix}$$

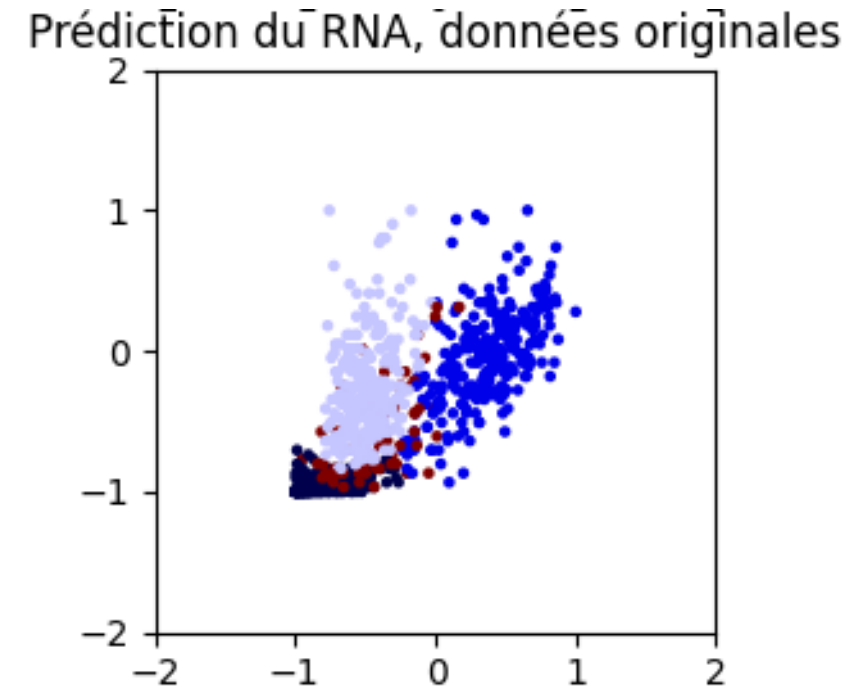
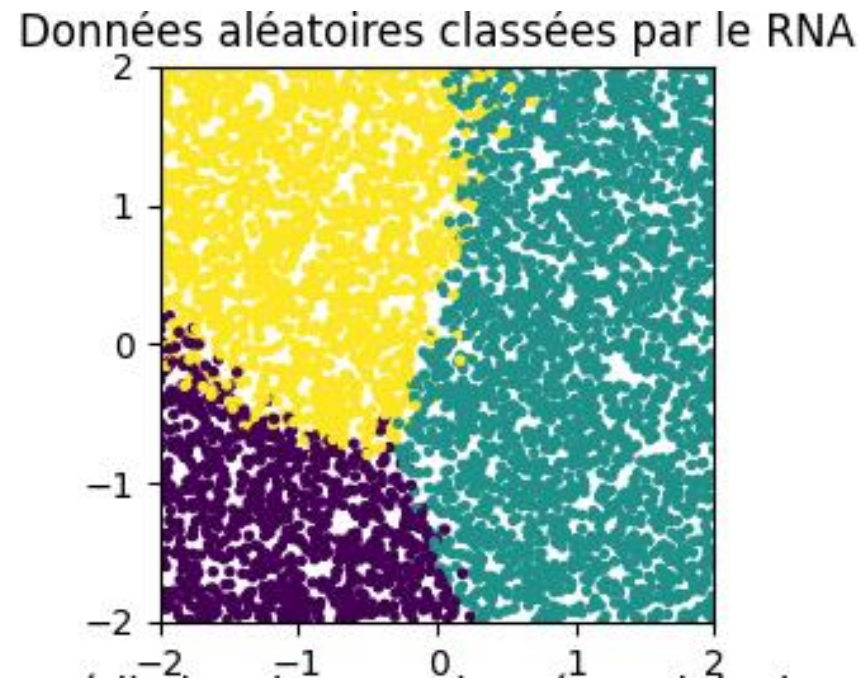
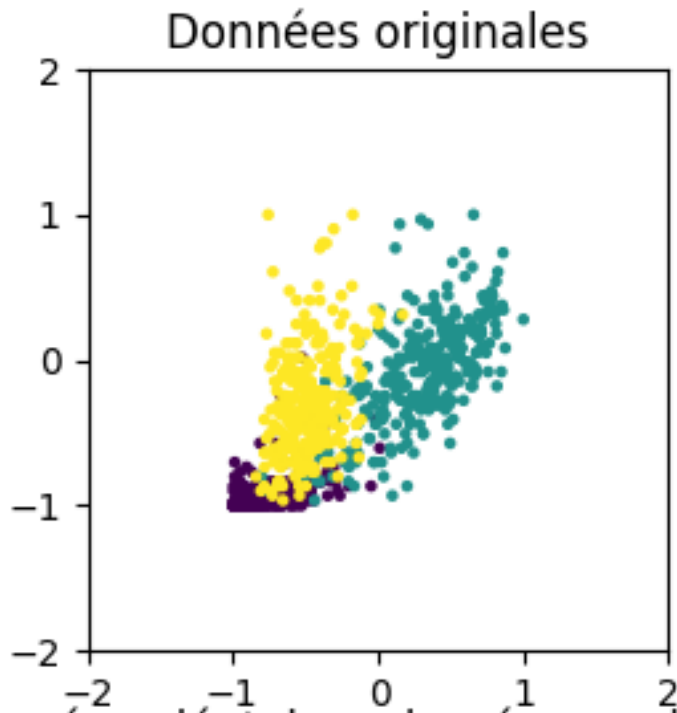
Prédiction du RNA, données originales



Classificateur Réseaux Neurons Artificielle

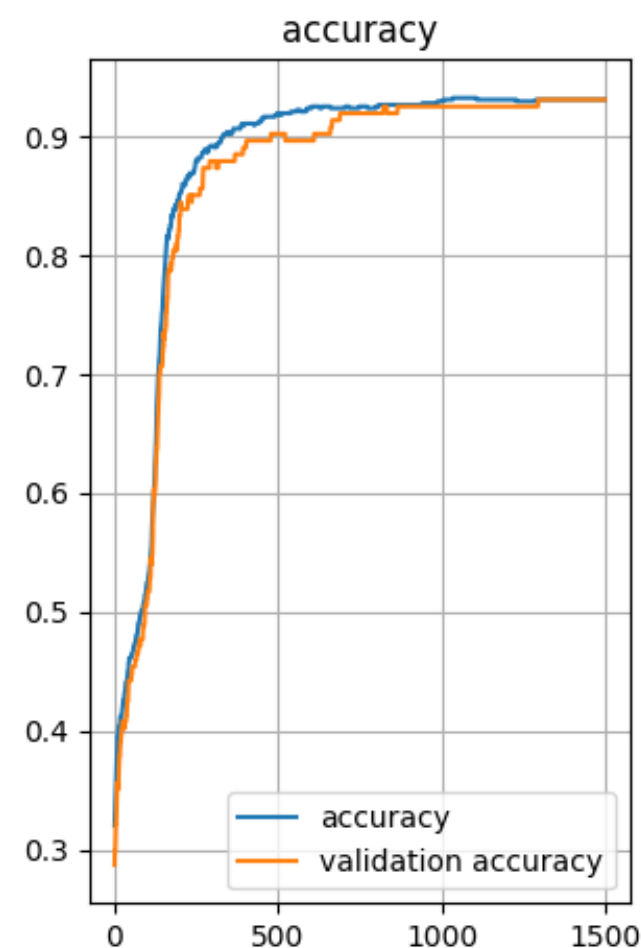
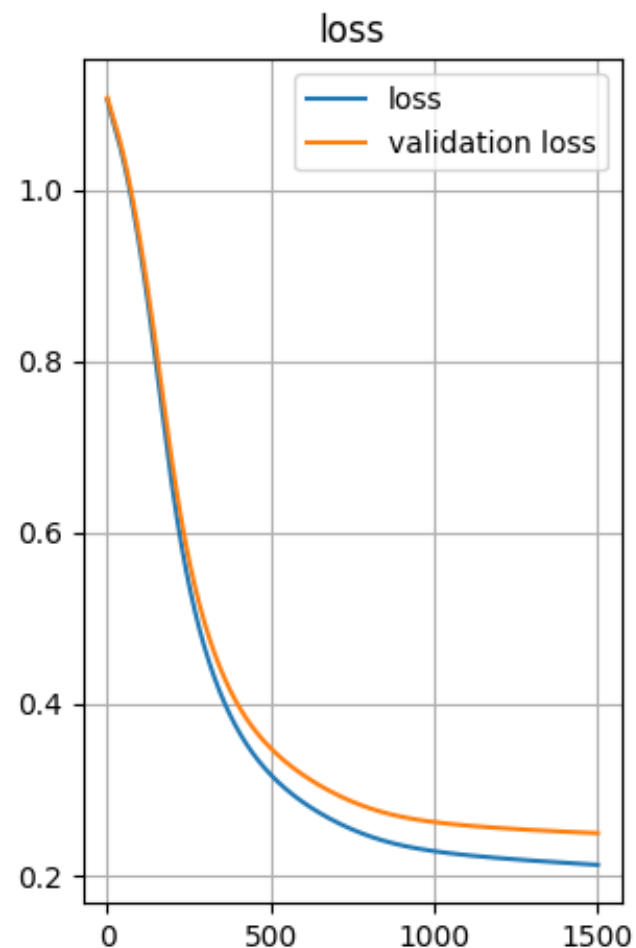
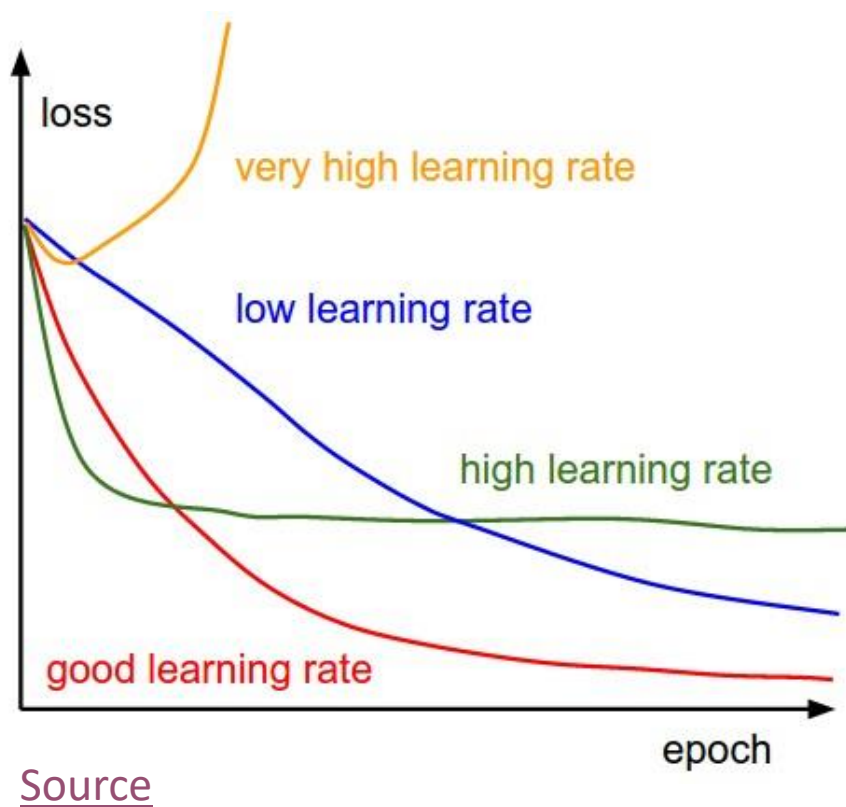
NN SimpleNN 9 layers cachées, 10 neurones par couche

- Plages
- Forêts
- Villes



Résultats

- *Bon taux d'apprentissage*
- *Exactitude très proche de l'entraînement*



Résultats

Type de Classificateur	Résultats (% d'erreur)	Matrice de confusion
Bayésien	7-8%	$\begin{bmatrix} 272 & 4 & 14 \\ 8 & 266 & 16 \\ 23 & 7 & 260 \end{bmatrix}$
K-PPV/Moy	10-11%	$\begin{bmatrix} 284 & 4 & 2 \\ 11 & 266 & 13 \\ 53 & 10 & 227 \end{bmatrix}$
RNA	6-8%	$\begin{bmatrix} 267 & 7 & 16 \\ 4 & 269 & 17 \\ 11 & 9 & 270 \end{bmatrix}$

Images en erreurs:

Plages

- Vert/autres couleurs (violet/bleu)
- Les images ayant des gros reflets
- Certains couchés de soleil sombre

Forets

- Pâle ou foncé

Villes

- bleuté et vert





Conclusion

- Représentation : hue et lignes dans l'image
- Niveau cible au-dessus de 90%
- Recommandations
 - Utilisation d'ensemble de données différents (validation et entraînement)
 - Avoir plus de données
 - Implémenter un taux d'apprentissage adaptatif
 - Observer la performance sur plus de 3 classes



Questions?


```
for object to mirror_mod.mirror_object
    operation == "MIRROR_X":
        mirror_mod.use_x = True
        mirror_mod.use_y = False
        mirror_mod.use_z = False
    operation == "MIRROR_Y":
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
    operation == "MIRROR_Z":
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True

    selection at the end -add
    mirror_ob.select= 1
    modifier_ob.select=1
    context.scene.objects.active
    ("Selected" + str(modifier_ob.name))
    mirror_ob.select = 0
    = bpy.context.selected_objects
    data.objects[one.name].select

print("please select exactly one object")

OPERATOR_CLASSES -----
```

Présentation du Code