

# Rendu du TP noté

## Exécution des programmes :

Nous avons plusieurs fichiers python :

- vocabulaire.py
- vectorisation.py
- apprentissage\_tests.py
- main\_recherche\_N\_2\_1\_2.py
- main\_vectorisation\_apprentissage\_3\_2.py
- main\_4\_1\_1.py
- main\_4\_1\_2.py
- main\_4\_2.py

L'ordre d'exécution de notre programme est le suivant :

- **main\_recherche\_N\_2\_1\_2.py** : pour créer les graphiques de la variance totale et de l'erreur max en fonction de N (prenant successivement les valeurs [2, 4, 8, 16, 32, 64, 128, 256, 512]). Ce script permet également de construire la matrice vocabulaire. Notre matrice sera construite sur la base de N=512 et enregistrée sous le nom '*matrice\_vocabulaire\_512\_clusters.txt*'.
- **main\_vectorisation\_apprentissage\_3\_2.py** : Pour vectoriser le résultat de l'extraction des SURF des images de nos dossiers d'apprentissage. Cette vectorisation est enregistrée dans un fichier nommé '*base\_512\_clusters.pickle*'.
- **main\_4\_1\_1.py** : Applique la version kernelisée de la LDA sur nos données d'apprentissage, en variant entre un degré de polynôme égal à 2 puis à 4. Le programme affiche le résultat de la transformation ainsi que la moyenne et la variance.
- **main\_4\_1\_2.py** : Crée un modèle SVC polynomial, l'entraîne avec deux classes de nos données d'apprentissage (éléphant et léopards) et affiche le résultat de la prédiction à partir de nos images de tests.
- **main\_4\_2.py** : Nous n'avons pas pu finir cette partie, cependant nous avons voulu tester de nouveau la classification par SVC polynomial en entraînant notre modèle avec nos 4 classes. Ce programme affiche le résultat de la classification de nos 16 images de tests sur ce modèle.

## 1 Consignes. Environnement de travail

### 1.3 Base d'images

Quelle est la majeure différence entre la plupart des classes de la base Caltech et celles que vous avez utilisées lorsque vous aviez appariés des descripteurs SURF entre eux ? Quelle incidence cela risque t'il d'avoir sur les résultats ?

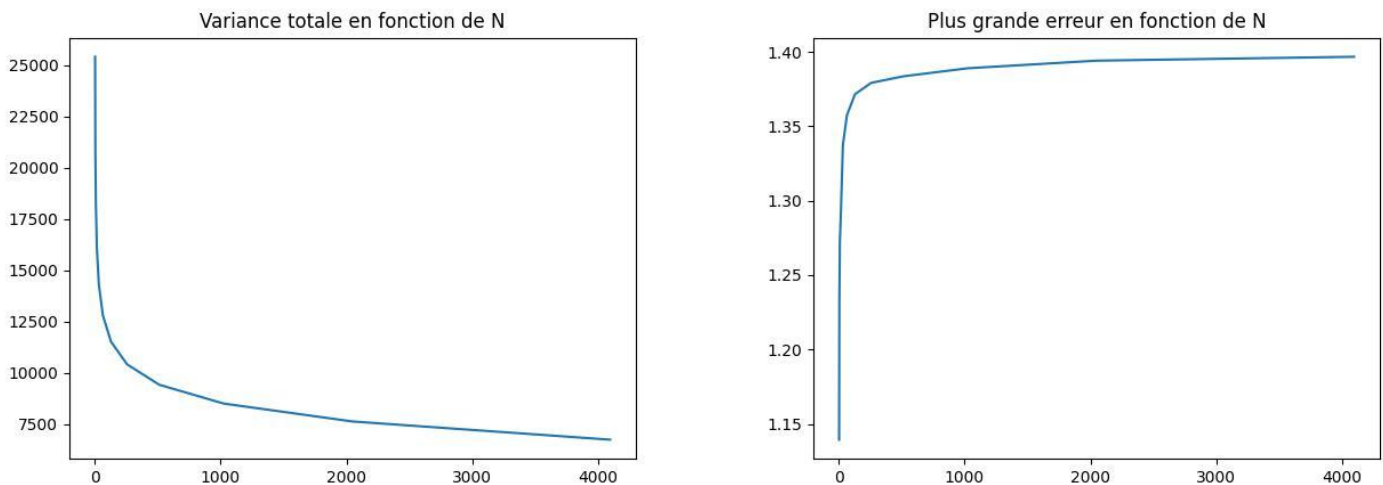
Lors du TP effectué en classe, nous avons des classes dont les images présentaient toujours le même sujet mais sous des angles de vue différents. L'algorithme SURF se chargeait donc de retrouver les descripteurs communs à chaque image.

Ici, les sujets de chaque image de chaque classe sont tous différents et même par leur nature : photographie, croquis... Les descripteurs communs entre les images seront alors beaucoup plus difficilement retrouvés par l'algorithme et les résultats seront donc probablement moins bons.

## 2 Choix d'images

### 2.1.2 Recherche de N

Nous avons choisi d'utiliser les classes "elephant", "wild\_cat", "panda" et "leopards" pour ce TP. Puis, nous obtenons les graphiques suivants :



Nous remarquons que notre graphique de la liste des erreurs maximum en fonction de N n'est pas cohérent : en effet, on s'attend à une diminution du nombre d'erreur maximum en fonction de l'augmentation du nombre de cluster N. Nous allons donc considérer l'inverse de cette courbe (donc ayant un visuel proche de celui de la liste des variances totales en fonction de N).

Nous observons donc qu'aux environs de 512 clusters la variance n'évolue plus aussi nettement que précédemment et que parallèlement les plus grandes erreurs ont atteint leur plafond bas de 1,15.

Nous allons donc conserver 512 comme valeur pour N dans la suite de ce TP.

A supposer que la méthode de classification que vous utilisiez après vectorisation soit la LDA, et que vous n'avez que 2 classes à séparer, quel impact pourrait avoir un vocabulaire visuel qui contiendrait trop de mots ? Pas assez de mots ?

Avec trop de mots, l'exécution de LDA prendrait beaucoup de temps de calcul. De plus, le programme risquerait de mal classer une image dans sa classe correspondante si elle ne présente pas tous les mots du vocabulaire visuel associé à cette classe. Avec pas assez de mots, la prédiction effectuée risque d'être mauvaise.

## 4 Apprentissage et tests

### 4.1.1 Effet de la Kernel-LDA

Nous avons utilisé les classes “elephant” et “leopards” pour cette partie.

A quoi correspondent les valeurs renvoyées par `s.transform()` ? Quel effet l'augmentation du degré du polynôme a-t-elle ?

Les valeurs renvoyées par `s.transform` correspondent au degré d'appartenance d'une image à une classe. L'augmentation du degré du polynôme permet de rendre encore plus précis ce degré d'appartenance en augmentant d'autant plus les valeurs. La séparation des données se verra ainsi améliorée, plus précise.

Essayez la même approche sur vos données `x` et `y` extraites en section 4.1, en faisant varier le degré si nécessaire. Commentez vos résultats : les projections semblent-elles indiquer une bonne séparation des classes ou non ? (la réponse peut être oui comme non, tout dépend des classes que vous avez choisies).

Puisque nous avons choisis des classes qui ne se ressemblent pas, nous nous attendons à avoir une bonne séparation des classes. C'est bien ce que nous obtenons et cela aussi en faisant varier le degré du polynôme.

Voici un extrait de résultat de `s.transform(X)` avec un polynôme de degré 2 :

Les images correspondantes à la classe “elephant” sont traitées en premier et leurs valeurs apparaissent comme positives alors que celles correspondantes à la classe “leopards” sont traitées plus tard et leurs valeurs sont négatives.

```
[....
[ 1.34392000e+01]
[ 1.04586185e+01]
[ 1.00322963e+01]
[ 1.77993883e+01]
[ 4.96089162e+00]
[ 7.92070577e+00]
[ 3.41608667e+03]
[ 1.58053773e+01]
[ 1.27029337e+01]
[ 2.23499370e+01]
...
[-7.73140969e-01]
[-7.60612674e-01]
[-6.10888930e-01]
[-6.98695884e-01]
[-1.49484889e+00]
[-4.16475451e-01]
[-4.66822844e-01]
[-4.77280730e-01]
[-1.42768790e+00]
[-1.84590773e+00]
[-1.88153494e-01]
...]
```

```
moyenne m1 : 197.54434504944615
moyenne m2 : -0.5235683021947939
variance v1 : 384604.0225766592
variance v2 : 0.5724207580495985
```

Voici un extrait de résultat de `s.transform(X)` avec un polynôme de degré 4 :

Les images correspondantes à la classe “elephant” sont traitées en premier et leurs valeurs apparaissent comme globalement positives et à l’échelle  $\times 10^3$  alors que celles correspondantes à la classe “leopards” sont traitées plus tard et leurs valeurs sont cette fois positives mais à l’échelle  $\times 10^1$ .

```
[...]
[ 1.03317864e+04]
[ 1.17110162e+04]
[ 2.30338388e+03]
[ 1.73657284e+03]
[ 1.08412403e+03]
[ 3.59836304e+03]
[ 5.49988744e+02]
[ 1.14788178e+03]
[-4.07907253e+06]
[ 1.92735296e+03]
[ 1.72649593e+03]
[ 5.66874829e+03]
...
[ 6.00043436e+01]
[ 7.59794638e+01]
[ 5.91053414e+01]
[ 8.66788171e+01]
[ 4.32541503e+01]
[ 1.09135702e+02]
[ 1.10482716e+02]
[ 1.04588577e+02]
[ 4.99339084e+01]
[ 2.75776219e+01]
[ 1.22022130e+02]
[ 1.37194205e+02]
[ 7.16205027e+01]
[ 1.34658788e+02]
[ 4.29451281e+01]
[ 3.83793491e+01]
...]
```

```
moyenne m1 : -145050.5617822299
moyenne m2 : 92.64425043655658
variance v1 : 390228034736.3424
variance v2 : 1618.9223385587952
```

#### 4.1.2 Séparation par SVC

Reportez vos résultats.

Nous avons effectué l'entraînement avec seulement 2 classes :

- classe 0 = elephant
- classe 1 : leopards

Nous avons construit notre modèle avec les paramètres suivant :

- kernel ='poly'
- degré de polynôme = 3
- constante de régularisation = 1.0

Et nous avons obtenu le résultat de prédiction suivant : [0 0 0 0 1 1 1 1 1 1 1 1 0 0 1 1]

Les elephant ont été prédit comme des elephant. Nos résultats correspondent à l'attendu.

Les leopards ont été prédit comme des leopards. Nos résultats correspondent à l'attendu.

Les panda ont été prédit comme des leopards. Cela était attendu puisque aucun entraînement n'a été fait sur des images de panda et que ce sujet ne ressemble ni à un éléphant, ni à un léopard.

Les wild\_cat ont été prédit comme des leopards et elephant. Cela était attendu puisque aucun entraînement n'a été fait sur des images de wild\_cat mais un peu surprenant puisque ce sujet ressemble à un léopard.

```

---- image : ./classes-caltech/elephant/test/image_0006.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/elephant/test/image_0017.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/elephant/test/image_0027.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/elephant/test/image_0046.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0001.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0172.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0177.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0198.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/panda/test/image_0001.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/panda/test/image_0008.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/panda/test/image_0032.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/panda/test/image_0038.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/wild_cat/test/image_0002.jpg ---- prediction : elephant ---- correct : False
---- image : ./classes-caltech/wild_cat/test/image_0005.jpg ---- prediction : elephant ---- correct : False
---- image : ./classes-caltech/wild_cat/test/image_0027.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/wild_cat/test/image_0034.jpg ---- prediction : Leopards ---- correct : False
---- Nombre de succès : 8
---- Nombre d'erreur : 8
---- taux de succès : 50.0

```

## 4.2 Apprentissage des quatres classes

Reportez les classifications obtenues pour chaque images de test non encore vue après les avoir vectorisées.

Nous avons cette fois-ci effectué l'entraînement avec les 4 classes :

Avec 4 clusters et kernel = 'rbf' :

```

---- image : ./classes-caltech/elephant/test/image_0046.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/panda/test/image_0008.jpg ---- prediction : elephant ---- correct : False
---- image : ./classes-caltech/wild_cat/test/image_0027.jpg ---- prediction : elephant ---- correct : False
---- image : ./classes-caltech/wild_cat/test/image_0002.jpg ---- prediction : elephant ---- correct : False
---- image : ./classes-caltech/Leopards/test/image_0177.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/panda/test/image_0001.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/elephant/test/image_0017.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/wild_cat/test/image_0034.jpg ---- prediction : elephant ---- correct : False
---- image : ./classes-caltech/elephant/test/image_0027.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/panda/test/image_0032.jpg ---- prediction : elephant ---- correct : False
---- image : ./classes-caltech/elephant/test/image_0006.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0001.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0172.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/wild_cat/test/image_0005.jpg ---- prediction : wild_cat ---- correct : True
---- image : ./classes-caltech/panda/test/image_0038.jpg ---- prediction : panda ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0198.jpg ---- prediction : Leopards ---- correct : True
---- Nombre de succès : 10
---- Nombre d'erreur : 6
---- taux de succès : 62.5

```

Avec 4 clusters et en changeant les paramètres pour kernel = 'poly' et degré de polynôme = 3 et constante de régularisation = 1, on obtient cette fois :

```

---- image : ./classes-caltech/wild_cat/test/image_0034.jpg ---- prediction : elephant ---- correct : False
---- image : ./classes-caltech/Leopards/test/image_0172.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/wild_cat/test/image_0002.jpg ---- prediction : wild_cat ---- correct : True
---- image : ./classes-caltech/wild_cat/test/image_0027.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/elephant/test/image_0017.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/panda/test/image_0038.jpg ---- prediction : panda ---- correct : True
---- image : ./classes-caltech/panda/test/image_0008.jpg ---- prediction : elephant ---- correct : False
---- image : ./classes-caltech/Leopards/test/image_0001.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/elephant/test/image_0027.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/elephant/test/image_0046.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/panda/test/image_0032.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/Leopards/test/image_0198.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0177.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/panda/test/image_0001.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/wild_cat/test/image_0005.jpg ---- prediction : wild_cat ---- correct : True
---- image : ./classes-caltech/elephant/test/image_0006.jpg ---- prediction : elephant ---- correct : True
---- Nombre de succès : 11
---- Nombre d'erreur : 5
---- taux de succès : 68.75

```

Le taux de réussite est légèrement supérieur à 50% mais ce n'est pas satisfaisant. Essayons d'obtenir un meilleur résultat en prenant un nombre de cluster adéquat.

Avec 512 clusters et kernel = 'rbf' :

```

---- image : ./classes-caltech/elephant/test/image_0006.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/elephant/test/image_0046.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/panda/test/image_0001.jpg ---- prediction : panda ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0001.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/wild_cat/test/image_0034.jpg ---- prediction : wild_cat ---- correct : True
---- image : ./classes-caltech/wild_cat/test/image_0027.jpg ---- prediction : wild_cat ---- correct : True
---- image : ./classes-caltech/panda/test/image_0008.jpg ---- prediction : panda ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0198.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/elephant/test/image_0017.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/panda/test/image_0032.jpg ---- prediction : panda ---- correct : True
---- image : ./classes-caltech/elephant/test/image_0027.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/panda/test/image_0038.jpg ---- prediction : panda ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0177.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0172.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/wild_cat/test/image_0005.jpg ---- prediction : wild_cat ---- correct : True
---- image : ./classes-caltech/wild_cat/test/image_0002.jpg ---- prediction : wild_cat ---- correct : True
---- Nombre de succès : 16
---- Nombre d'erreur : 0
---- taux de succès : 100.0

```

Le taux de réussite est de 100% : cette fois-ci toutes les prédictions sont justes.

En changeant les paramètres pour kernel = 'poly' et degré de polynôme = 3 et constante de régularisation = 1, on obtient cette fois :

```

---- image : ./classes-caltech/Leopards/test/image_0177.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/wild_cat/test/image_0027.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/elephant/test/image_0046.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/elephant/test/image_0006.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0198.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/panda/test/image_0038.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/elephant/test/image_0017.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/elephant/test/image_0027.jpg ---- prediction : elephant ---- correct : True
---- image : ./classes-caltech/Leopards/test/image_0172.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/panda/test/image_0032.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/panda/test/image_0008.jpg ---- prediction : panda ---- correct : True
---- image : ./classes-caltech/panda/test/image_0001.jpg ---- prediction : Leopards ---- correct : False
---- image : ./classes-caltech/Leopards/test/image_0001.jpg ---- prediction : Leopards ---- correct : True
---- image : ./classes-caltech/wild_cat/test/image_0005.jpg ---- prediction : wild_cat ---- correct : True
---- image : ./classes-caltech/wild_cat/test/image_0002.jpg ---- prediction : wild_cat ---- correct : True
---- image : ./classes-caltech/wild_cat/test/image_0034.jpg ---- prediction : wild_cat ---- correct : True
---- Nombre de succès : 12
---- Nombre d'erreur : 4
---- taux de succès : 75.0

```

Le taux de réussite est de 75%. Les résultats sont assez satisfaisants bien que les prédictions sur la classe panda ne le sont pas.

## 5 Bonus

Est-il envisageable d'aboutir à une méthode de classification similaire ? Si oui, quelles modifications proposez-vous d'apporter ? Sinon, pourquoi est-il impossible d'intégrer cette nouvelle information dans les décisions ?

Nous pensons qu'en plus de SURF, ces nouvelles informations peuvent améliorer la robustesse des descripteurs et permettre par la suite d'obtenir de meilleures prédictions.

Au lieu d'être simplement recherché sur  $x$  et  $y$ , un descripteur prendrait en plus les coordonnées couleurs  $r, g, b$ . Il s'agirait alors de descripteurs IVC (Image Vers Couleur). Ces coordonnées couleurs pourraient être ensuite comparées à une gamme de coordonnées couleurs attendues et permettraient enfin d'effectuer un filtrage pour ne garder que les descripteurs d'intérêt. Les classifications en résultant devraient alors donner de meilleurs résultats.