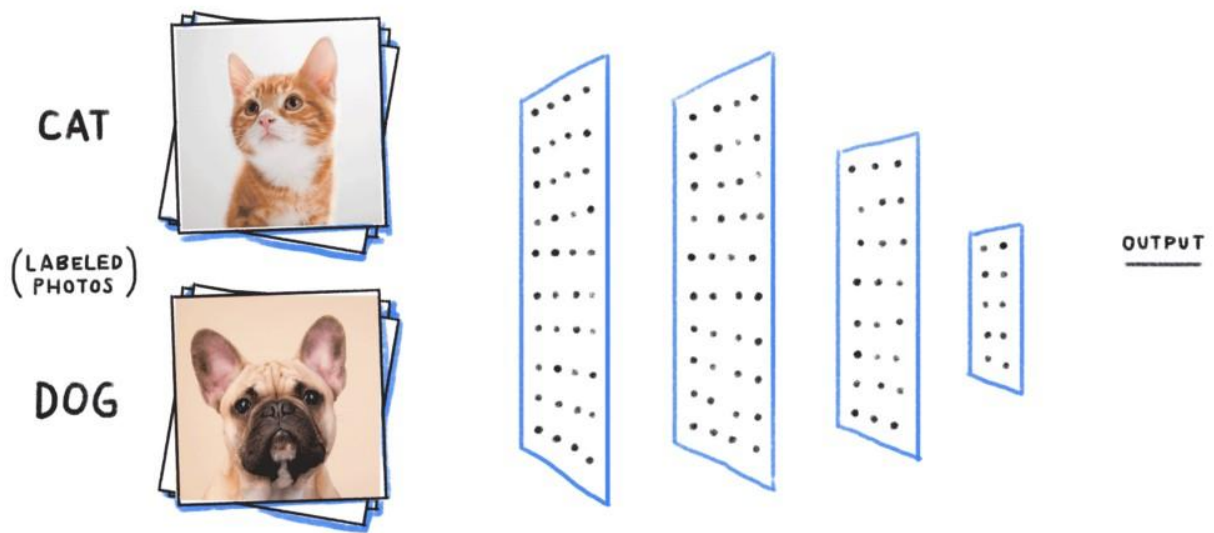


Projet de classification d'images

ACI



Maël le Vot

11/04/2021

ESIR 2

INTRODUCTION

Ce projet de classification d'image va nous permettre d'appliquer nos connaissances sur les principaux classifieurs étudiés en cours et en TP, et d'expérimenter les méthodes apprises en utilisant notre propre jeu de données choisi au préalable.

Il s'agira donc d'extraire les descripteurs que nous appliquerons à nos images et à les utiliser par la suite pour entraîner nos classifieurs.

Dans notre cas, nous allons expérimenter l'application des descripteurs Histogramme couleurs (RGB) et SIFT (Scale-Invariant Feature Transform) sur les classifieurs : Arbre de décisions et Random Forest.

Enfin, voici une description des différents fichiers utilisés et de leur utilité :

- dataset (folder) : dossier contenant les données (toutes les images)
- datas.txt : fichier contenant l'id de toutes les données ainsi que leur classe respective
- descripteurs_histoRGB.txt : fichier contenant le résultat de l'extraction du descripteur Histogramme RGB pour chaque image ainsi que la classe de chacun des éléments
- descripteurs_SIFT.txt : fichier contenant le résultat de l'extraction du descripteur SIFT pour chaque image ainsi que la classe de chacun des éléments
- histoRGB_descriptor.py : script python executant l'extraction du descripteur Histogramme RGB pour chaque image
- SIFT_descriptor.py : script python exécutant l'extraction du descripteur SIFT pour chaque image
- pre_processing.py : script python ayant pour but de remplir le fichier datas.txt
- script_projet.py : script python appelant les deux scripts d'extraction de descripteurs afin de structurer les données dans les fichier texte descripteurs_histoRGB.txt et descripteurs_SIFT.txt
- script_projet.R : script R exécutant les commandes relatives à l'entraînement des classifieurs et aux différentes expérimentation : script à lancer pour visualiser les résultats obtenus

DESCRIPTION DU JEU DE DONNÉES

Descripteur : Histogramme RGB

Nombre d'exemples : 1123

Nombre d'attributs : 768 (= 3×256 valeurs d'histogramme)

Nombre de classes : 4 (cloudy, rain, shine et sunrise)

Répartition des classes :

Classe	cloudy	rain	shine	sunrise
Nombre d'éléments	298	215	253	357

Construction du jeu d'apprentissage : 70% du jeu de données comme étudié en TP (ici 786 exemples)

Construction du jeu de validation (Arbre de décisions) : 15% du jeu de données (ici 168 exemples)

Construction du jeu de test (Arbre de décisions) : 15% restants (ici 169 exemples)

Construction du jeu de test (Random Forest) : 30% restants (ici 337 exemples)

Descripteur : SIFT

Nombre d'exemples : 1102

Nombre d'attributs : 1280 (= 10×128 valeurs, c'est à dire 128 valeurs par points d'intérêts)

Nombre de classes : 4 (cloudy, rain, shine et sunrise)

Répartition des classes :

Classe	cloudy	rain	shine	sunrise
Nombre d'éléments	298	215	253	336

Le descripteur SIFT se base sur la détection de points clés dans chaque image à partir de l'estimation du gradient dans plusieurs zones. Il en ressort un vecteur de 128 valeurs par point clés servant de descripteur de la zone en question.

Le problème est que le descripteur SIFT ne trouve pas le même nombre de points clés pour toutes les images, alors que pour notre problème de classification, il nous faut le même nombre d'attributs pour chaque exemple. C'est pourquoi nous décidons de ne garder que 10 points clés par image, afin d'avoir tout de même 1280 attributs par exemple, soit près de deux fois plus que pour les histogrammes RGB, mais aussi pour pouvoir conserver un nombre d'exemples raisonnable.

En effet, le descripteur détecte moins de 10 points clés dans certaines images, c'est pourquoi nous n'utiliserons pas ces dernières dans nos expériences. Choisir un nombre de points clés plus élevé aurait entraîné une diminution de notre jeu de données encore plus importante, c'est la raison pour laquelle nous nous contenterons de 10.

Construction du jeu d'apprentissage : 70% du jeu de données comme étudié en TP (ici 771 exemples)

Construction du jeu de validation (Arbre de décisions) : 15% du jeu de données (ici 165 exemples)

Construction du jeu de test (Arbre de décisions) : 15% restants (ici 166 exemples)

Construction du jeu de test (Random Forest) : 30% restants (ici 331 exemples)

APPRENTISSAGE PAR ARBRE DE DÉCISION

Descripteur : Histogramme RGB

Pour la construction de l'arbre T_{max} , on utilise la fonction `rpart()` de R en utilisant l'ensemble d'apprentissage. On peut ensuite déterminer le nombre de prédictions correctes que cet arbre effectuera sur notre jeu de test en utilisant la fonction `predict()`. On s'attend à obtenir un taux de bonnes classifications élevé du fait que l'arbre ne soit pas encore élagué.



On obtient donc le taux suivant : $Taux = \frac{110}{169} \simeq 0.65$

Comme nous l'attendions, le taux de bonnes classifications est relativement élevé (65%)

d'exemples bien classés). Pour mieux se représenter ce résultat, nous pouvons le mettre sous la forme d'une matrice de confusion (Figure 1).

Prédiction → Vérité	cloudy	rain	shine	sunrise
cloudy	21	23	7	2
rain	7	16	4	1
shine	6	1	29	1
sunrise	4	3	0	44

Figure 1 : Matrice de confusion sur le jeu de test (histogramme RGB)

 : bonnes classifications
 : mauvaises classifications

Grâce à cette matrice, on remarque que le cas pour lequel le modèle se trompe le plus de fois est lorsqu'il confond la classe cloudy avec la classe rain (il classe beaucoup d'images dans la classe rain alors qu'elles appartiennent à la classe cloudy (23), ce qui s'explique par le fait que toutes les images de pluies possèdent des nuages. Il est donc probable que cette erreur ne fasse qu'augmenter lorsque nous élaguerons l'arbre.

Maintenant que nous avons nos premiers résultats, nous pouvons nous occuper de l'élagage de l'arbre. Pour cela, nous utilisons les fonctions vues en TP : *rev()*, qui nous permet de récupérer les valeurs de *cp* à mettre dans la fonction *prune()*, elle-même servant à élaguer l'arbre construit précédemment.

Ici, la fonction *rev()* nous donne 17 valeurs de *cp* à utiliser. Nous allons donc élaguer 17 fois l'arbre de départ *Tmax* à partir de ces valeurs et observer le nombre de prédictions correctes effectuées sur l'ensemble de validation, afin de choisir l'arbre étant à la fois suffisamment performant tout en étant moins complexe que l'arbre *Tmax*. On obtient donc les résultats de la Figure 2 :

cp	Bonnes prédictions	Bonnes prédictions
----	--------------------	--------------------

	ensemble d'apprentissage	ensemble de validation
0	786	117
0.0009398496	780	118
0.001879699	756	119
0.002349624	751	119
0.002819549	748	119
0.003759398	711	118
0.004699248	706	119
0.005639098	694	119
0.007518797	642	116
0.009398496	622	120
0.01503759	590	116
0.03383459	572	111
0.04887218	546	107
0.05827068	484	97
0.06954887	447	93
0.1240602	381	74
0.2387218	254	52
Nombre d'éléments dans l'ensemble	786	168
Figure 2 : Tableau des bonnes classifications sur arbres élagués (histogramme RGB)		

L'arbre que l'on choisit est donc celui correspondant à $cp = 0.00939398496$ car il génère le meilleur taux de bonnes classifications sur l'ensemble de validation :

$$Taux = \frac{120}{168} \simeq 0.71$$

On peut donc maintenant, à partir de cet arbre, calculer l'erreur de généralisation sur l'ensemble de test. On obtient alors :

$$Erreur = \frac{60}{169} \simeq 0.35$$

Ce résultat est satisfaisant étant donné que pour l'arbre T_{max}, nous obtenions une erreur quasi identique ($\frac{59}{169} \simeq 0.349$), alors qu'ici, l'arbre est beaucoup moins complexe que T_{max}.

Descripteur SIFT

Pour commencer, nous obtenons le taux de bonnes classifications sur l'ensemble de test avec l'arbre T_{max} suivant :

$$Taux = \frac{57}{166} \simeq 0.34$$



On remarque que ce taux est très faible et peut s'expliquer par deux raisons :

1. Le modèle de l'arbre de décision n'est pas performant sur un jeu de données de trop grande taille (hypothèse peu probable)
2. Le descripteur utilisé n'est pas performant (le choix de n'utiliser que 10 points clés ne permet pas d'avoir de très bons résultats)

Si nous mettons ce résultat sous la forme d'une matrice de confusion, on obtient la Figure 3 :

Prédiction → Vérité	cloudy	rain	shine	sunrise
cloudy	17	12	9	11
rain	4	8	8	12
shine	4	11	8	14
sunrise	2	12	10	24

Figure 3 : Matrice de confusion sur le jeu de test (SIFT)

 : bonnes classifications
 : mauvaises classifications

Malgré ces résultats, nous pouvons tout de même tenter de déterminer le l'arbre le plus performant en élaguant Tmax. Pour cela on effectue à nouveau les opérations appliquées pour les histogrammes RGB, et on obtient le tableau Figure 4 :

cp	Bonnes prédictions ensemble d'apprentissage	Bonnes prédictions ensemble de validation
0	770	56
0.001901141	715	51
0.003802281	641	52
0.004752852	636	52
0.005703422	603	57
0.006337136	585	58
0.007604563	551	61
0.008238276	538	57
0.008555133	524	58
0.00887199	490	53
0.009505703	475	53
0.01140684	463	52
0.01330798	428	58
0.0161597	367	50
0.01711027	358	50
0.01901141	328	52
0.02661597	314	51
0.06558935	245	43

Nombre d'éléments dans l'ensemble	771	165
<i>Figure 4 : Tableau des bonnes classifications sur arbres élagués (SIFT)</i>		

L'arbre que l'on choisit est donc celui correspondant à $cp = 0.007604563$ car il génère le meilleur taux de bonnes classifications sur l'ensemble de validation :

$$Taux = \frac{61}{165} \simeq 0.37$$

On peut donc maintenant, à partir de cet arbre, calculer l'erreur de généralisation sur l'ensemble de test. On obtient alors :

$$Erreur = \frac{120}{166} \simeq 0.72$$

Le résultat montre qu'avec un arbre élagué, le descripteur SIFT s'en sort moins bien qu'avec Tmax ($Erreur Tmax = \frac{109}{166} \simeq 0.66$)

Cela peut s'expliquer par le fait que l'arbre soit beaucoup moins complexe, et donc de ce fait, moins précis, mais aussi car comme remarqué au début de la partie, le descripteur SIFT utilisé dans ces conditions (nombre limité de points clés) ne permet pas d'obtenir de résultats convaincants en classification d'image.

APPRENTISSAGE PAR RANDOM FOREST

Descripteur : Histogramme RGB



Pour commencer, rappelons qu'une Random Forest se construit en prenant environ 66% de l'ensemble d'apprentissage avec remise pour construire chaque arbre, ce qui signifie que chaque arbre sera différent car construit aléatoirement. Le reste de l'ensemble pour chaque arbre (environ 33%) sont les données OOB (Out Of Bag) qui serviront à approximer l'erreur réelle. Nous verrons que nous pourrions être plus précis en utilisant un jeu de test.

- **Forêt avec un seul arbre**

Dans un premier temps, nous allons générer une forêt contenant un seul et unique arbre. On obtient alors la matrice de confusion suivante (Figure 5) avec en plus l'erreur générée par classe :

Prédiction → Vérité	cloudy	rain	shine	sunrise	Erreur par classe
cloudy	30	24	11	5	0.2542
rain	11	27	6	2	0.3529
shine	16	8	42	8	0.2289
sunrise	4	2	4	78	0.1299

Figure 5 : Matrice de confusion sur l'ensemble OOB (Histo RGB)

 : bonnes classifications
 : mauvaises classifications

À partir de là, on peut calculer une approximation de l'erreur réelle et on obtient la valeur suivante :

$$E_{estimée} = \frac{101}{278} \simeq 0.3633$$

Cette première approximation est équivalente à l'erreur de généralisation trouvée dans le cas des arbres de décisions ($\simeq 0.35$) ce qui peut déjà nous montrer que cette méthode de classification pour un arbre est aussi efficace.

On peut maintenant calculer l'erreur empirique (sur l'ensemble d'apprentissage) et l'erreur réelle (sur l'ensemble de test). On obtient les résultats suivants :

$$E_{empirique} \simeq 0.1285$$

$$E_{réelle} \simeq 0.3027$$

L'erreur réelle correspond donc bien à l'estimation faite plus tôt et est légèrement inférieure à celle calculée pour un arbre de décision (tout en restant dans le même ordre de grandeur). Malgré cela, ce classifieur paraît plus performant même avec un seul arbre, ce qui peut s'expliquer par le fait qu'une Random Forest intègre une évaluation de l'erreur à chaque étape (erreur sur l'OOB), ce qui n'est pas le cas d'un arbre de décision classique, qui peut donc plus facilement devenir instable.

- **Forêt avec plusieurs arbres**

Afin d'estimer l'efficacité de ce classifieur, nous allons comparer les erreurs réelles

obtenues sur des forêts contenant entre 2 et 10 arbres. On obtient alors le tableau représentant ces résultats Figure 6 :

Nombre d'arbres	2	3	4	5	6	7	8	9	10
Erreur réelle	0.2671	0.2641	0.1632	0.1810	0.2136	0.2136	0.2136	0.1573	0.1781

Figure 6 : Erreurs réelles des forêts entre 2 et 10 arbres (Histo RGB)

Dans cet intervalle, l'erreur est minimisée quand la forêt contient 9 arbres. On arrive alors à une erreur réelle ≈ 0.1573 et donc, à un taux de bonnes classifications ≈ 0.8427 . Ces résultats sont bien meilleurs que ceux obtenus avec un arbre de décision simple, ce qui s'explique également par le fait que le classifieur Random Forest ne peut pas faire de sur-apprentissage en mode bagging, ce qui nous permet de multiplier les arbres sans craindre d'augmenter l'erreur réelle.



Descripteur : SIFT

- Forêt avec un seul arbre

Pour une forêt ne contenant qu'un seul arbre, on obtient la matrice de confusion Figure 7 :

Prédiction → Vérité	cloudy	rain	shine	sunrise	Erreur par classe
cloudy	23	8	17	34	0.2542
rain	10	11	16	14	0.3529
shine	13	20	16	22	0.2289
sunrise	12	17	19	38	0.1299

Figure 7 : Matrice de confusion sur l'ensemble OOB (SIFT)

 : bonnes classifications
 : mauvaises classifications

À partir de là, on peut calculer l'approximation de l'erreur réelle et on obtient la valeur

suivante :

$$E_{\text{estimée}} = \frac{202}{290} \simeq 0.6966$$

C'est une erreur très élevée même pour une forêt à un arbre, mais comme nous l'avions expliqué pour les arbres de décisions, le descripteur SIFT utilisé comme ceci avec seulement 10 points clés retenus n'obtient pas de bons résultats de classification.

On peut tout de même calculer l'erreur empirique (sur l'ensemble d'apprentissage) et l'erreur réelle (sur l'ensemble de test). On obtient les résultats suivants :

$$E_{\text{empirique}} \simeq 0.2619$$

$$E_{\text{réelle}} \simeq 0.6858$$

On remarque que l'erreur empirique reste acceptable mais que l'erreur réelle est bien fidèle à l'estimation réalisée précédemment. Toutefois, elle demeure légèrement plus faible que celle obtenue avec un arbre de décision classique ($\simeq 0.72$), ce qui démontre encore une fois l'efficacité de la méthode Random Forest.

- **Forêt avec plusieurs arbres**

De la même manière que pour le descripteur Histogramme RGB, nous allons comparer les erreurs réelles obtenues sur des forêts contenant entre 2 et 10 arbres. On obtient alors le tableau représentant ces résultats Figure 8 :

Nombre d'arbres	2	3	4	5	6	7	8	9	10
Erreur réelle	0.6828	0.6405	0.6314	0.5831	0.5952	0.5589	0.5438	0.5409	0.5075

Figure 8 : Erreurs réelles des forêts entre 2 et 10 arbres (SIFT)

Comme on peut le remarquer, l'erreur ne fait que décroître au fur et à mesure que l'on augmente le nombre d'arbres, jusqu'à arriver à une erreur d'environ 50%. On se dit donc qu'en augmentant encore davantage le nombre d'arbres nous pourrions arriver à un taux de bonnes classifications acceptable. Nous allons donc effectuer la même opération pour un nombre d'arbres prenant les valeurs 30, 50 et 100 (Figure 9) :

Nombre d'arbres	30	50	100
Erreur réelle	0.4229	0.4078	0.3444

Figure 9 : Erreurs réelles des forêts entre 30 et 100 arbres (SIFT)

Comme nous pouvons le voir, il nous aura fallu une forêt de 100 arbres pour obtenir des résultats satisfaisants avec le descripteur SIFT, c'est à dire une erreur réelle de :

$$Er_{réelle} \simeq 0.3444$$

C'est une erreur acceptable dans la mesure où elle avoisine celle obtenue avec le descripteur Histogramme RGB sur les arbres de décisions ($\simeq 0.35$)

CONCLUSION

Pour conclure ce projet, il serait intéressant de visualiser les principaux résultats obtenus sous la forme d'un tableau récapitulatif (Figure 10) :

	Descripteur Histo RGB	Descripteur SIFT
Arbre de décision	Bonnes classifications : 65%	Bonnes classifications : 28%
Random Forest	Bonnes classifications : 84.27% (9 arbres)	Bonnes classifications : 65.56% (100 arbres)

Figure 10 : Tableau récapitulatif des expériences effectuées

Comme nous l'attendions, c'est le descripteur Histogramme RGB sur le classifieur Random Forest qui obtient les meilleurs résultats.

Cela s'explique par le fait que ce descripteurs renvoie toujours le même nombre de valeurs descriptives de l'image, là où le descripteur SIFT ne renvoie pas le même nombre de points d'intérêts par image. De plus, le fait d'avoir forcé l'obtention de 10 points clés par image a sûrement joué sur les résultats.

Malgré cela, on obtient des résultats assez convaincants pour ce descripteur sur une

Random Forest à condition de générer beaucoup d'arbres (ici 100).

Au final, ce projet m'aura permis de comprendre toute la chaîne de fabrication d'un système de classification d'images, en partant de l'extraction des descripteurs (pas toujours évident surtout pour le format des données), jusqu'à l'entraînement des classifieurs. Cela a donc été très instructif.