

Réalisez des indexations automatiques  
d'images





# Agenda

- 1 Contexte
- 2 Approche
- 3 Présentation des jeux de données
- 4 Analyse SIFT
- 5 Analyse CNN
- 6 Conclusion
- 7 Prochaines étapes
- 8 Annexe : l'environnement technique



# 1) Contexte

## Objectif

La base de données d'une association de protection des animaux commence à grandir et ils n'ont pas toujours le temps de référencer les images des animaux accumulées depuis des années. Ils aimeraient donc réaliser un index de l'ensemble de la base de données d'images qu'ils possèdent, pour classer les chiens par races.

## Données mises à disposition

Le Stanford Dogs Dataset

## Mission

Réalisation d'un algorithme de détection de la race du chien sur base d'une photo, afin d'accélérer leur travail d'indexation.



## 2) Approche



### 1) Présentation

Présentation du jeu de données (Nombre de photos, Répartition des photos par race, analyse des hauteurs et largeurs des images...), Travail préparatoire (renommage des noms de répertoires, organisation de différents jeux de train/test.)

### 2) Analyse SIFT



Sélection de l'algorithme, Nettoyage des images (Noir et Blanc, Whitening, Std, ...), Data Augmentation, MixedUp, ...



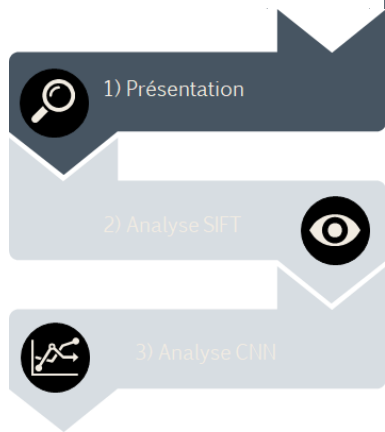
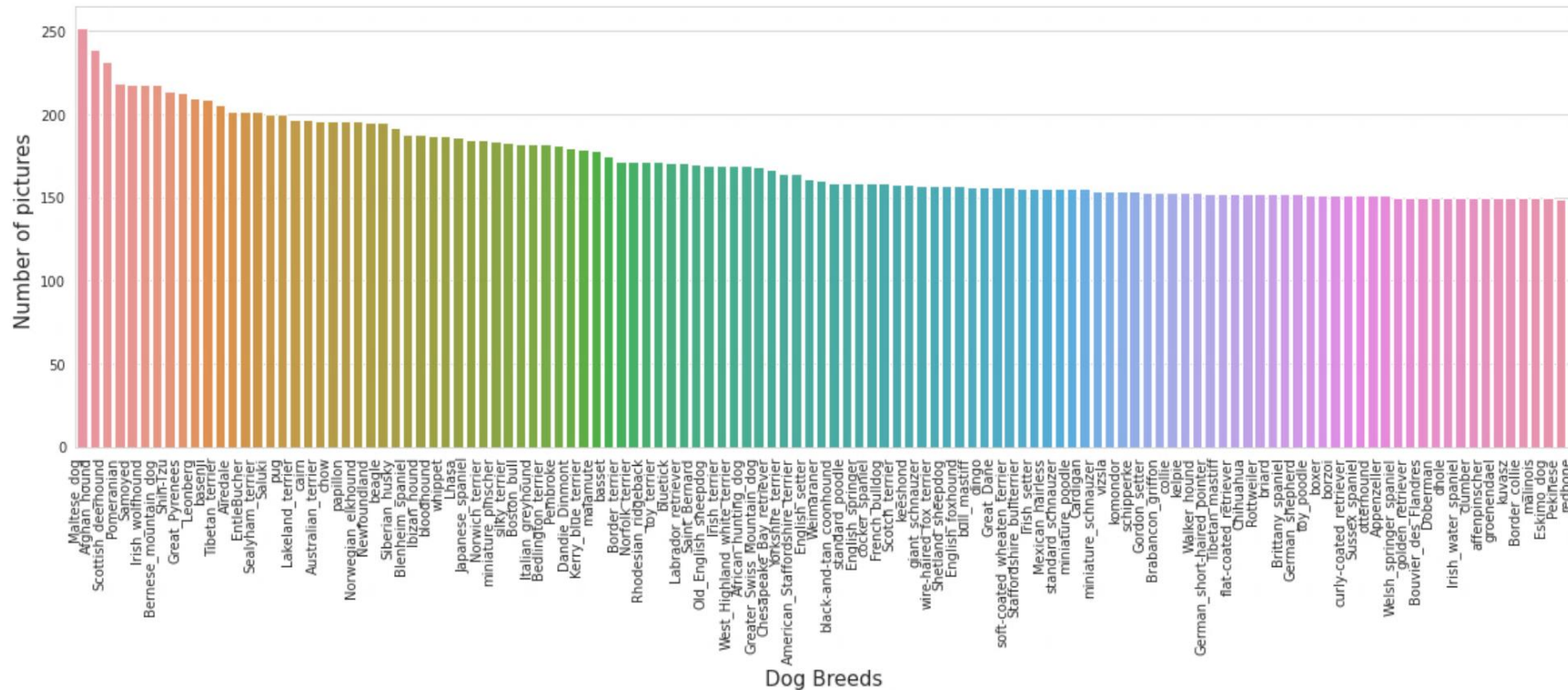
### 3) Analyse CNN

Nettoyage des images (Noir et Blanc, Whitening, Std, ), Data Augmentation, Mixed Up CNN from Scratch, CNN Transfer Learning



### 3) Présentation des jeux de données (1/3)

Nombre d'images par race





### 3) Présentation des jeux de données (2/3)



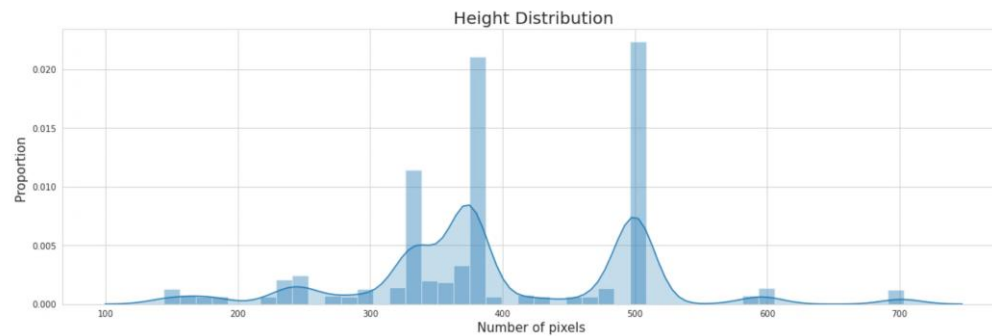
1) Présentation

2) Analyse SIFT



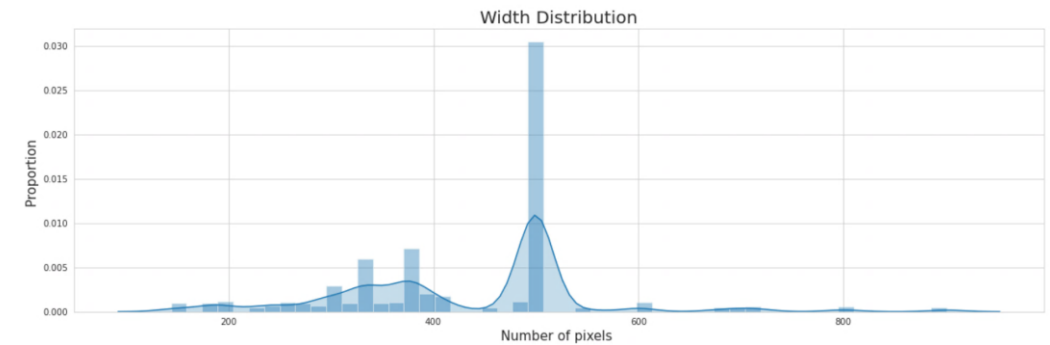
3) Analyse CNN

#### Analyse des hauteurs des images



Caractéristiques	Valeur
Nombre de lignes	20580
Moyenne du nombre de pixels	395.572109
Ecart type	101.737213
Min de valeurs	144.000000
25%	333.000000
50%	375.000000
75%	500.000000
Max	703.000000

#### Analyse des largeurs des images

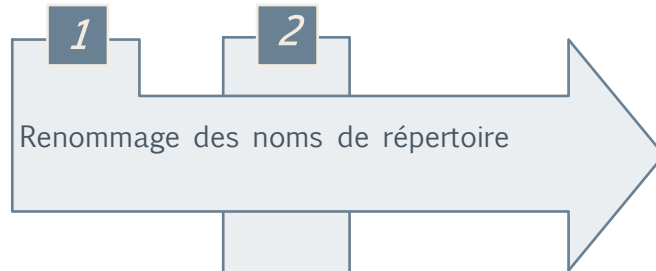


Caractéristiques	Valeur
Nombre de lignes	20580
Moyenne du nombre de pixels	432.038678
Ecart type	120.367525
Min de valeurs	144.000000
25%	340.000000
50%	500.000000
75%	500.000000
Max	901.000000







### 3) Présentation des jeux de données (3/3)

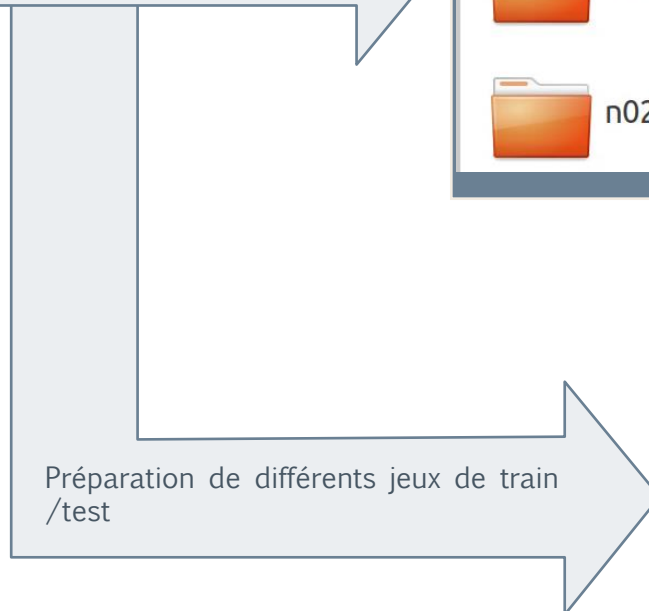
#### Travail préparatoire



Les changements opérés sont :

Name
 n02085620-Chihuahua
 n02085782-Japanese_spaniel

Name
 Chihuahua
 JapaneseSpaniel



#### Les jeux de tests suivantes ont été créés :

Nom	Full	Nom	Standard
Nombre de races	120	Nombre de races	12
Nombre d'images	20580	Nombre d'images	1382
Ratio	70% train / 30% test	Ratio	70% train / 30% test

Nom	Small	Nom	Tiny
Nombre de races	6	Nombre de races	3
Nombre d'images	962	Nombre d'images	498
Ratio	70% train / 30% test	Ratio	70% train / 30% test



1) Présentation

2) Analyse SIFT



3) Analyse CNN



## 4) Analyse SIFT

### Sélection de l'algorithme

Sur base de l'étude Image Matching Using SIFT, SURF, and ORB: Performance Comparison for Distorted Images :  
<https://arxiv.org/ftp/arxiv/papers/1710/1710.02726.pdf>



1) Présentation

2) Analyse SIFT



3) Analyse CHH

Domain	Criteria	SIFT	SURF	ORB
Varying Intensity	Time	↙	↗	↗
	Match Rate	↗	↗	→
Rotated Image	Time	↙	↗	↗
	Match Rate	↗	↗	→
Scaled Image	Time	↙	↗	↗
	Match Rate	→	→	↗
Sheared Image	Time	↙	↗	↗
	Match Rate	↗	↗	→

Légende :







# 4) Analyse SIFT

## Démarche SIFT

### 1. Préparation

#### 1.1 Redimensionnement des images

Pour chaque photo :

- Redimensionnement 224 X 224

#### 1.2 Pré-processing

Noir & Blanc, Whitening, Std, ..MixedUp

### 2. Extraction des features

#### 2.1 Extraction

Pour chaque image extraction de toutes les features

#### 2.2 Limitation à 150 points centraux par feature

Application d'un KMEANS sur chaque feature

### 3. Résultats

#### 3.1 Réduction à deux dimensions

Application d'une TSNE

#### 3.2 Prédiction de la classe pour les images de tests via Catboost

#### 3.3 Calcul de l'accuracy

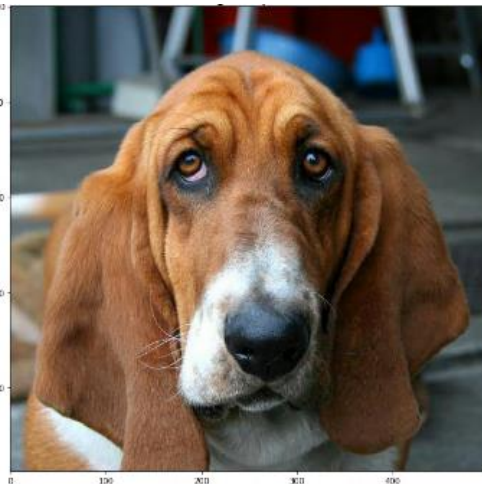


Image originale



Image nettoyée

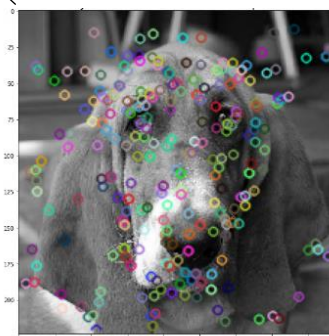


Image avec les features



1) Présentation

2) Analyse SIFT



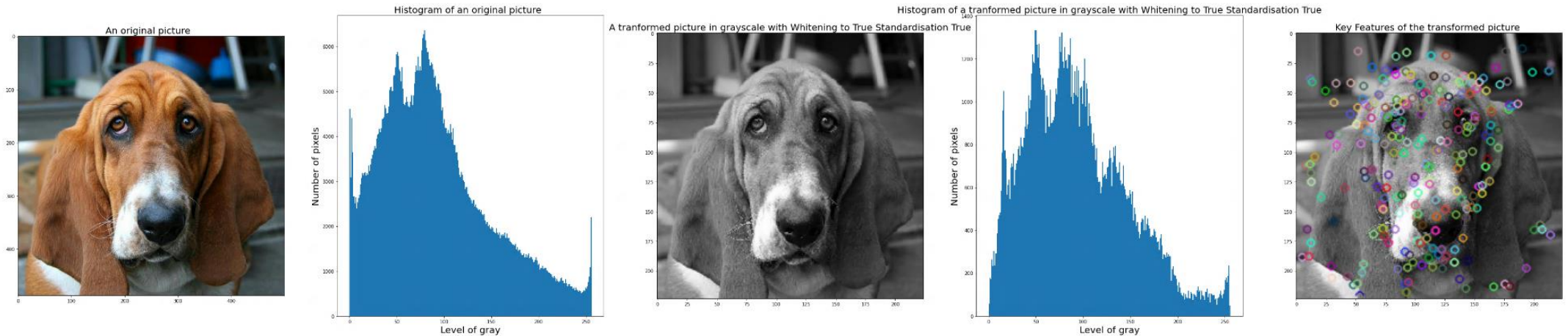
3) Analyse CHH



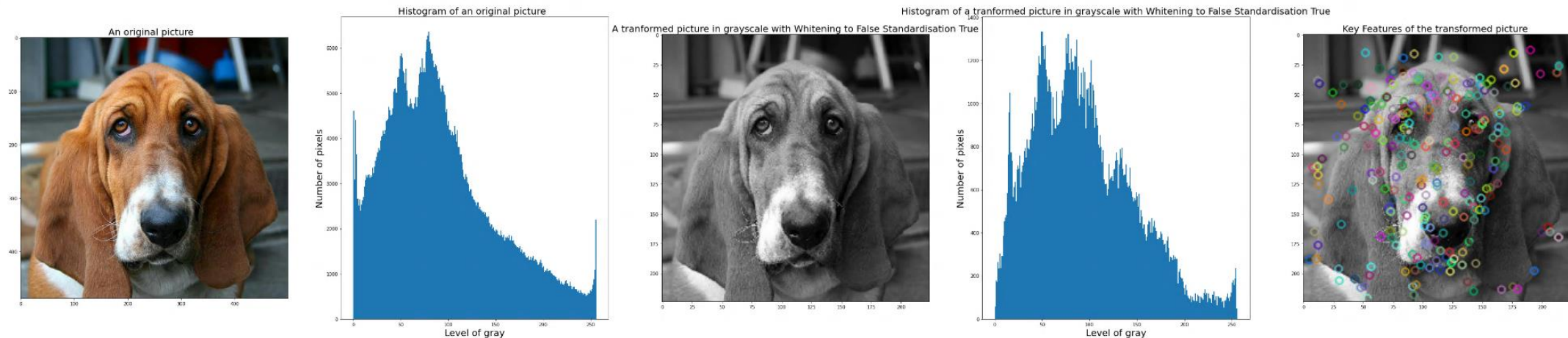
# 4) Analyse SIFT

Comparaison des méthodes de preprocessing (1/2)

1) Whitening : True / Std : True



2) Whitening : False / Std : True



1) Présentation

2) Analyse SIFT



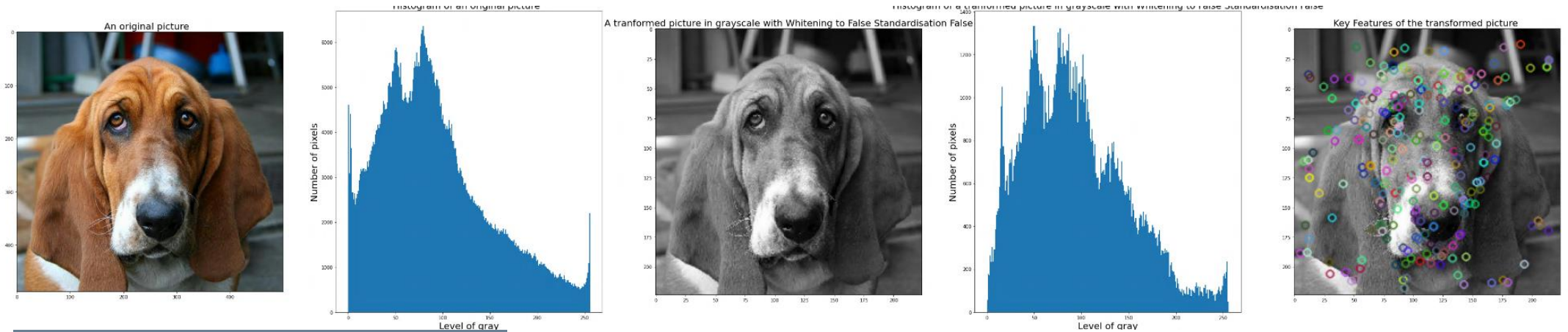
3) Analyse CHH



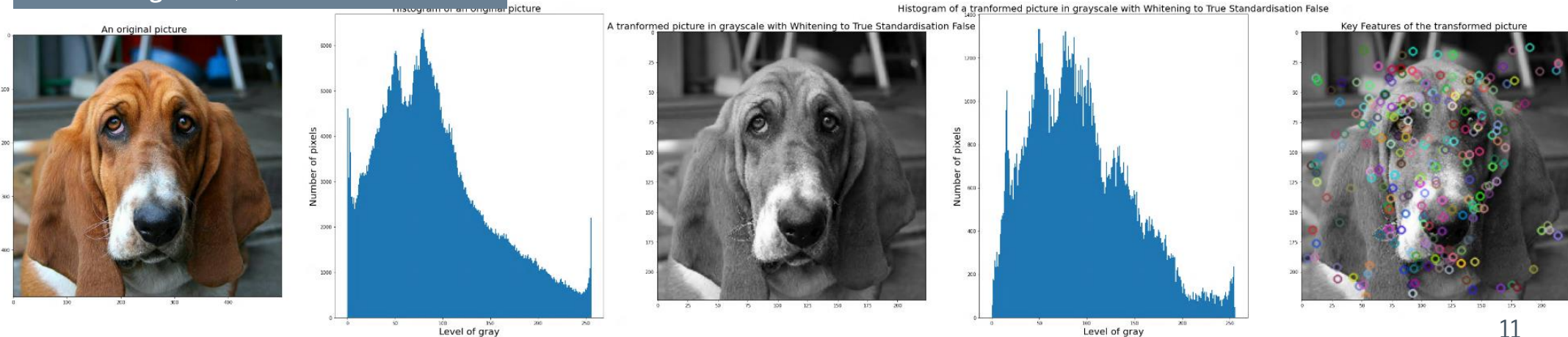
# 4) Analyse SIFT

Comparaison des méthodes de preprocessing (2/2)

3) Whitening : False / Std : False



4) Whitening : True / Std : False



1) Présentation

2) Analyse SIFT



3) Analyse CHH

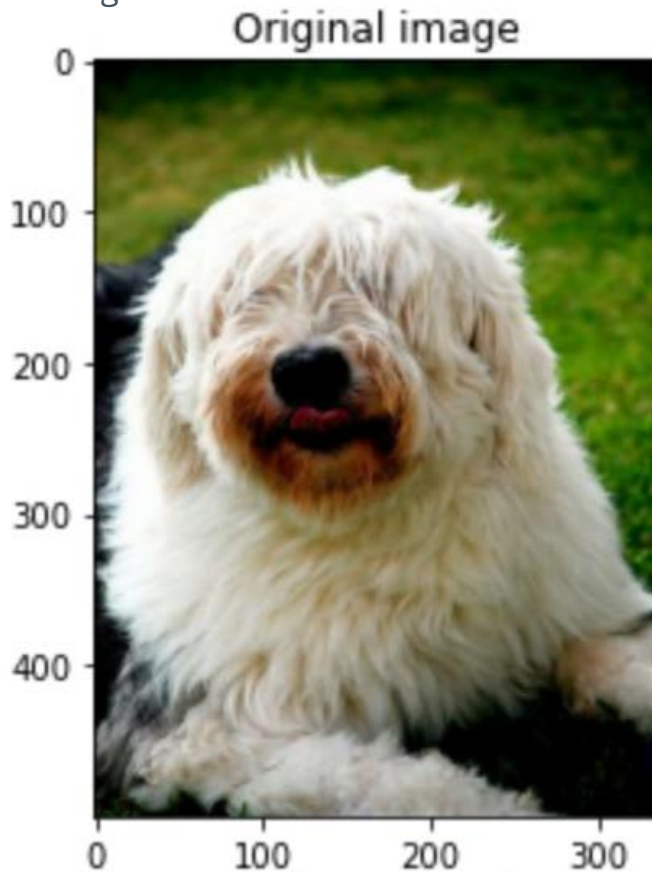




## 4) Analyse SIFT

### Data Augmentation

**Principe :** Génération de nouvelles images modifiées sur base d'images existantes (rotation, zoom, symétrie axiale, ....). Cependant, il y a des dépendances vis à vis du jeu de données pouvant mener à un sur-apprentissage.



1) Présentation

2) Analyse SIFT



3) Analyse CNN



## 4) Analyse SIFT

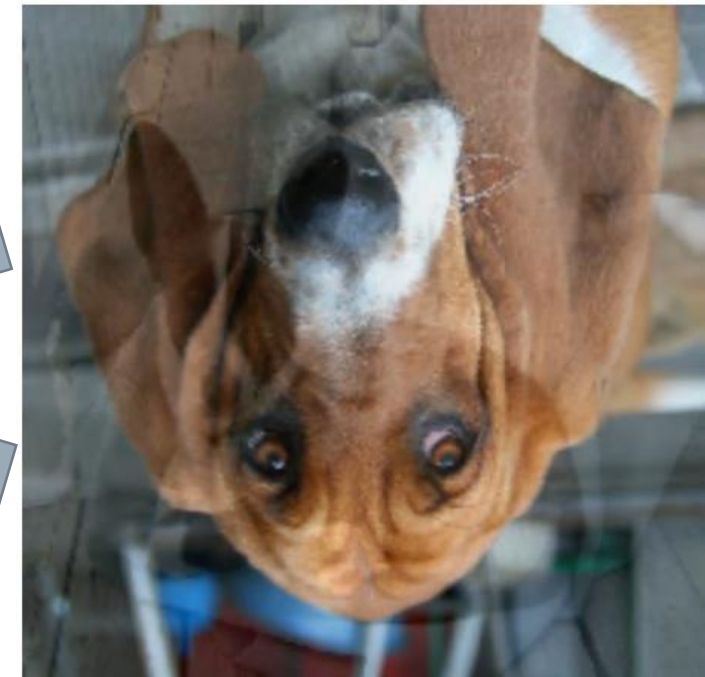
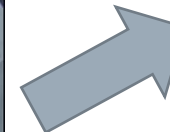
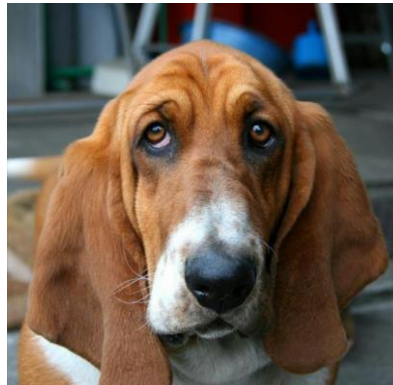
### Mixed Up

Principe : Description dans le papier : <https://arxiv.org/pdf/1710.09412.pdf>

L'objectif est de créer un dataset sur base d'images appartenant à des categories différentes



- Modélisation de relations entre individus de différentes categories
- Amélioration des modèles dans la prise de décisions



1) Présentation

2) Analyse SIFT



3) Analyse CHH



# 4) Analyse SIFT

Résultats obtenus sur le jeux de données Tiny



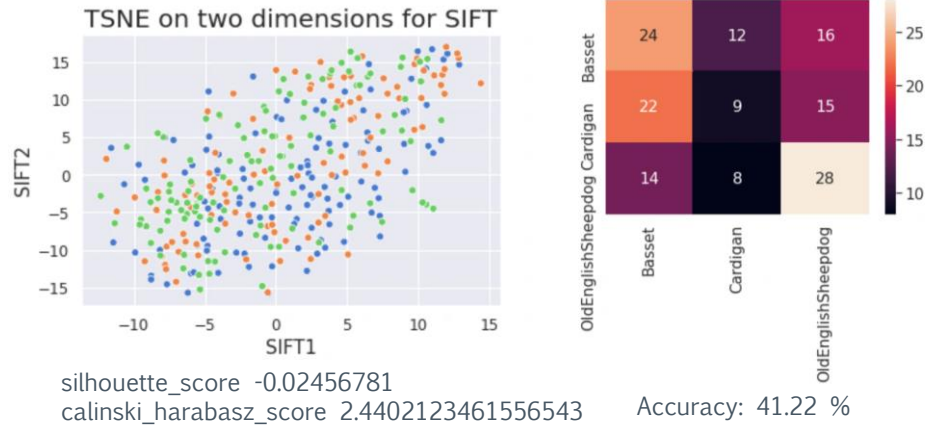
1) Présentation

2) Analyse SIFT

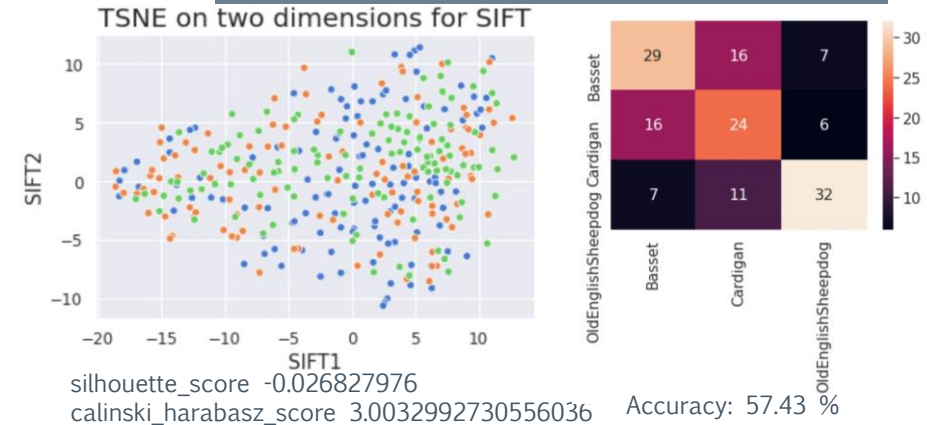


3) Analyse CHH

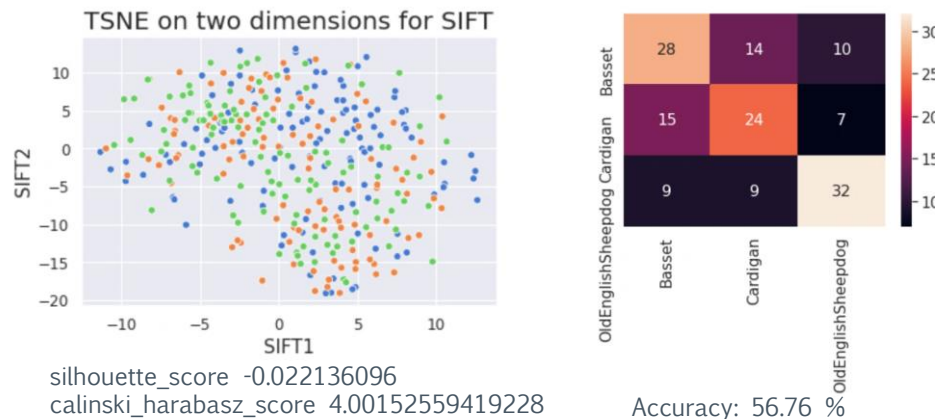
## 1) Whitening : True / Std : True



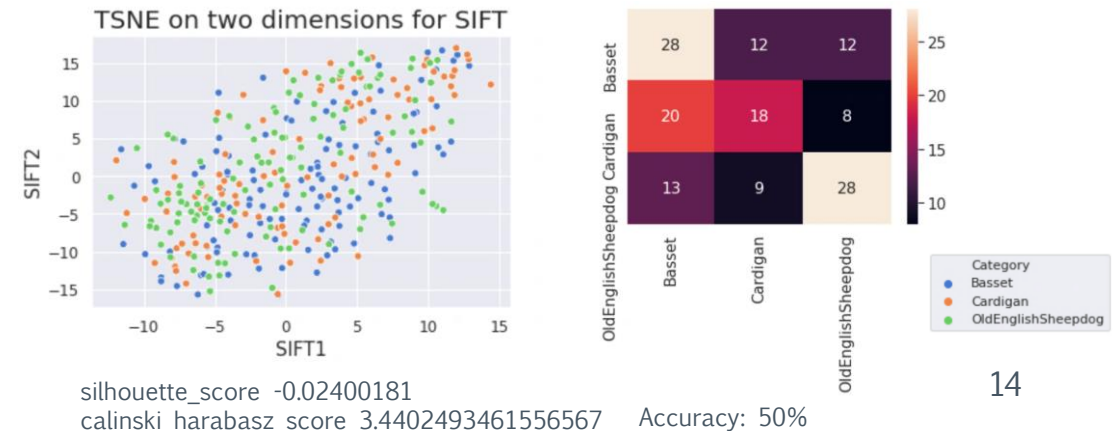
## 2) Whitening : False / Std : True



## 3) Whitening : False / Std : False



## 4) Whitening : True / Std : False





# 4) Analyse SIFT

Résultats obtenus sur le jeux de données Tiny



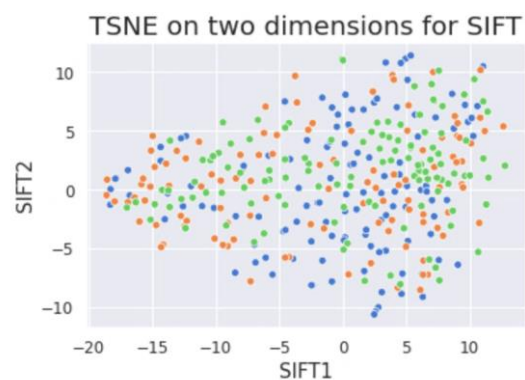
1) Présentation

2) Analyse SIFT

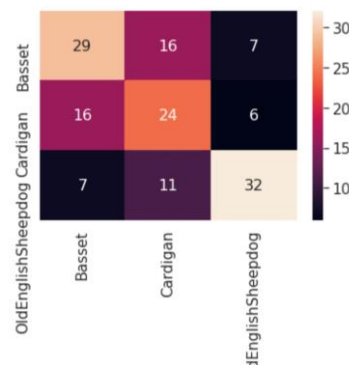


3) Analyse CHH

1) Whitening : False / Std : True

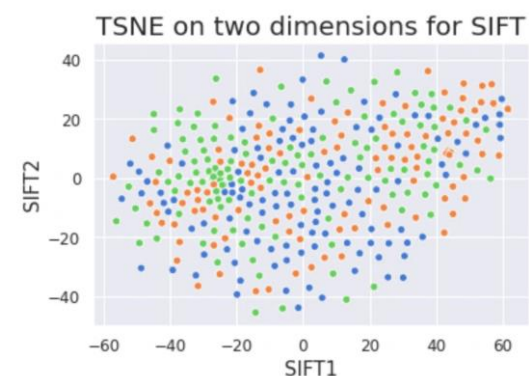


silhouette\_score -0.026827976  
calinski\_harabasz\_score 3.0032992730556036

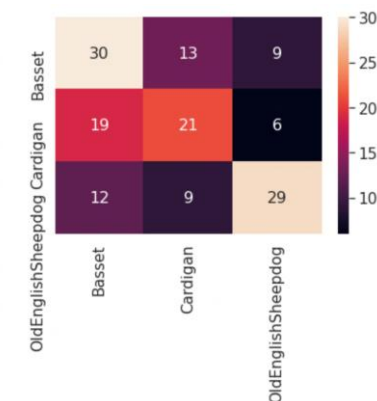


Accuracy: 57.43 %

2) Whitening : False / Std : True avec MixedUP



silhouette\_score -0.020029584  
calinski\_harabasz\_score 7.324503443115568 Accuracy: 54.05 %







# 5) Analyse CNN

## Démarche CNN from Scratch



1) Présentation

2) Analyse SIFT



3) Analyse CNN

### 1. Préparation

#### 1.1 Séparation Train / Test / Validation

Nom	Full
Nombre de races	120
Nombre d'images	20580
Ratio	70% train / 30% test Validation (20% du train)

Nom	Standard
Nombre de races	12
Nombre d'images	1382
Ratio	70% train / 30% test Validation (20% du train)

#### 1.2 Redimensionnement des images en 224 x 224

### 2. Entraînement du modèle

#### 2.1 Modèle

```
Model: "sequential"
-----
Layer (type)                Output Shape          Param #
-----
conv2d (Conv2D)              (None, 222, 222, 16)  448
max_pooling2d (MaxPooling2D) (None, 111, 111, 16)  0
conv2d_1 (Conv2D)             (None, 109, 109, 32)  4640
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 32)    0
conv2d_2 (Conv2D)             (None, 52, 52, 64)    18496
max_pooling2d_2 (MaxPooling2 (None, 26, 26, 64)    0
conv2d_3 (Conv2D)             (None, 24, 24, 128)   73856
max_pooling2d_3 (MaxPooling2 (None, 12, 12, 128)   0
conv2d_4 (Conv2D)             (None, 10, 10, 256)   295168
max_pooling2d_4 (MaxPooling2 (None, 5, 5, 256)    0
flatten (Flatten)             (None, 6400)          0
dropout (Dropout)             (None, 6400)          0
dense (Dense)                 (None, 512)           3277312
dense_1 (Dense)               (None, 128)           61560
-----
Total params: 3,731,480
Trainable params: 3,731,480
Non-trainable params: 0
```

#### 2.2 Optimizer

Adam est choisi car :

- Peu gourmand en mémoire
  - Tuning des hyperparamètres simplifié
- L'hyperparamètre learning\_rate est à 0.0001

### 3. Résultats

#### 3.1 Accuracy sur le jeu de tests de train et validation

#### 3.2 Accuracy sur le jeu de tests après entraînement

#### 3.3 Matrice de confusion et classification report





## 5) Analyse CNN

CNNFromScratch : StandardDataSet No Data Augmentation

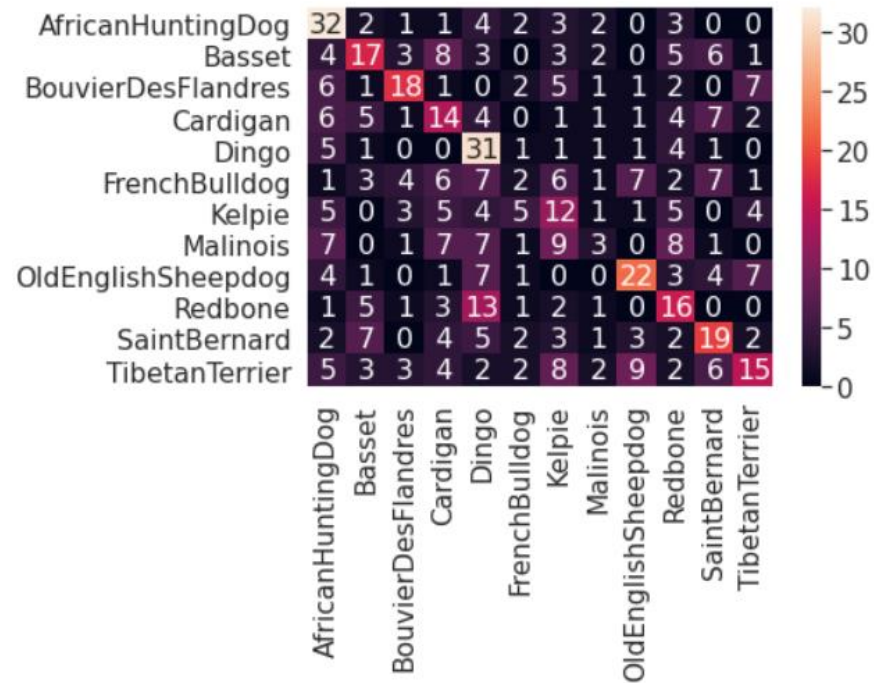


1) Présentation

2) Analyse SIFT



3) Analyse CNN



	precision	recall	f1-score	support
AfricanHuntingDog	0.41	0.64	0.50	50
Basset	0.38	0.33	0.35	52
BouvierDesFlandres	0.51	0.41	0.46	44
Cardigan	0.26	0.30	0.28	46
Dingo	0.36	0.67	0.47	46
FrenchBulldog	0.11	0.04	0.06	47
Kelpie	0.23	0.27	0.24	45
Malinois	0.19	0.07	0.10	44
OldEnglishSheepdog	0.49	0.44	0.46	50
Redbone	0.29	0.37	0.32	43
SaintBernard	0.37	0.38	0.38	50
TibetanTerrier	0.38	0.25	0.30	61
accuracy			0.35	578
macro avg	0.33	0.35	0.33	578
weighted avg	0.33	0.35	0.33	578



## 5) Analyse CNN

CNNFromScratch : StandardDataSet With Data Augmentation

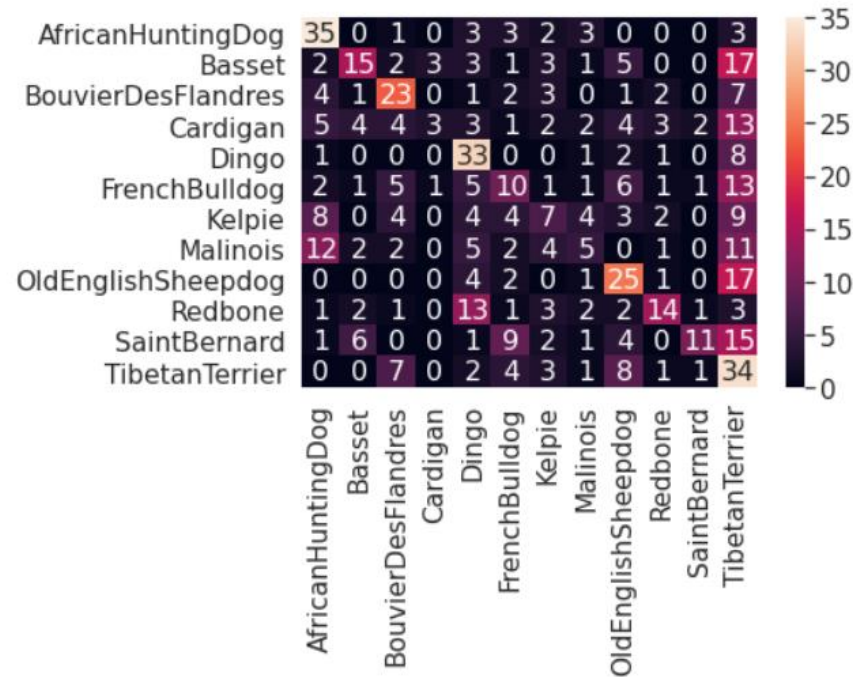


1) Presentation

2) Analyse SIFT



3) Analyse CNN



	precision	recall	f1-score	support
AfricanHuntingDog	0.49	0.70	0.58	50
Basset	0.48	0.29	0.36	52
BouvierDesFlandres	0.47	0.52	0.49	44
Cardigan	0.43	0.07	0.11	46
Dingo	0.43	0.72	0.54	46
FrenchBulldog	0.26	0.21	0.23	47
Kelpie	0.23	0.16	0.19	45
Malinois	0.23	0.11	0.15	44
OldEnglishSheepdog	0.42	0.50	0.45	50
Redbone	0.54	0.33	0.41	43
SaintBernard	0.69	0.22	0.33	50
TibetanTerrier	0.23	0.56	0.32	61
accuracy			0.37	578
macro avg	0.41	0.36	0.35	578
weighted avg	0.41	0.37	0.35	578



# 5) Analyse CNN

CNNFromScratch : FullDataSet



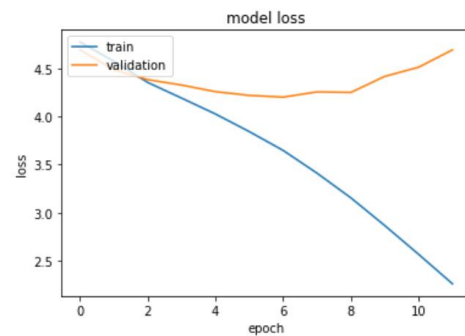
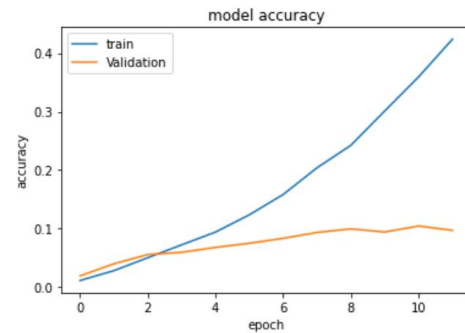
1) Présentation

2) Analyse SIFT



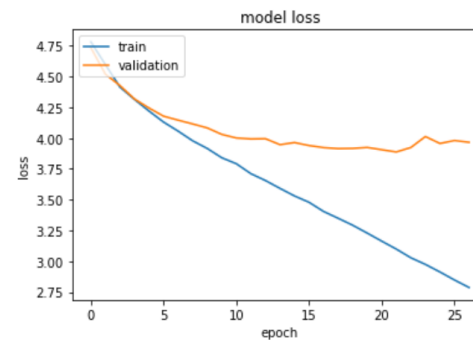
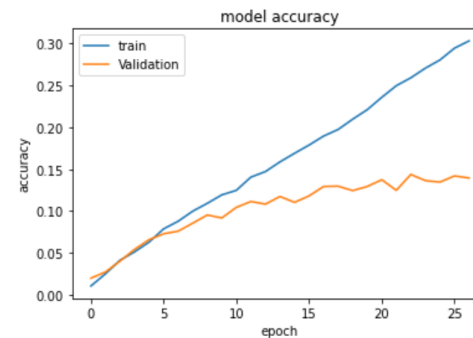
3) Analyse CNN

## 1) No Data Augmentation



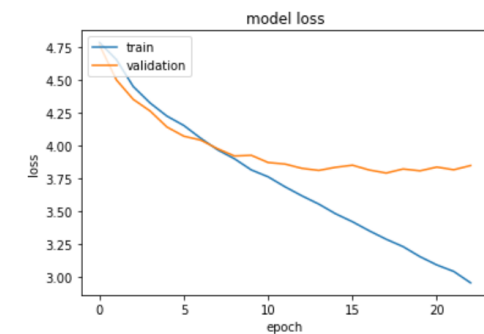
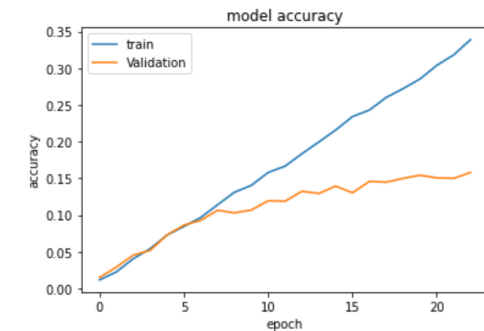
accuracy			0.09	6063
macro avg	0.08	0.09	0.08	6063
weighted avg	0.09	0.09	0.08	6063

## 2) Data Augmentation



accuracy			0.12	6063
macro avg	0.12	0.12	0.11	6063
weighted avg	0.12	0.12	0.11	6063

## 3) Mixed Up



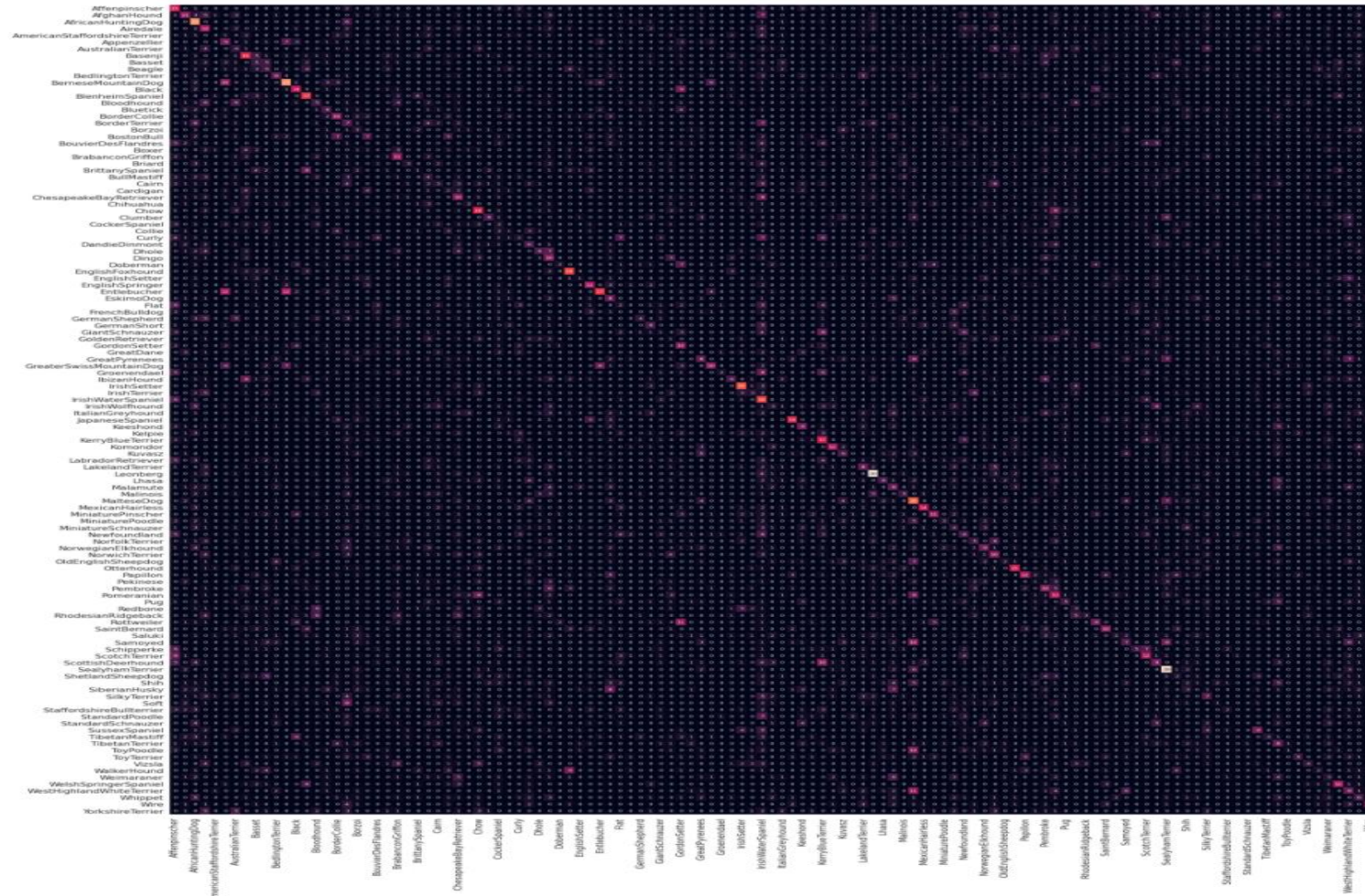
accuracy			0.14	6063
macro avg	0.15	0.14	0.13	6063
weighted avg	0.15	0.14	0.13	6063





# 5) Analyse CNN

CNNFromScratch : MixedUp Confusion Matrix



1) Présentation

2) Analyse SIFT

3) Analyse CNN



# 5) Analyse CNN

## Démarche CNN Transfer Learning



1) Présentation

2) Analyse SIFT



3) Analyse CNN

### 1. Préparation

#### 1.1 Séparation Train / Test / Validation

Nom	Full
Nombre de races	120
Nombre d'images	20580
Ratio	70% train / 30% test Validation (20% du train)

Nom	Standard
Nombre de races	12
Nombre d'images	1382
Ratio	70% train / 30% test Validation (20% du train)

#### 1.2 Redimensionnement des images en 224 x 224

### 2. Entraînement du modèle

#### 2.1 Modèles

- Xception est choisi pour son bon rapport performance / taille.
- EfficientB7 est choisi pour sa precision

Pour le Transfer Learning, l'approche "fine-tuning partielle a été choisie (**Entraînement du classifieur et des couches basses**),

#### 2.2 Optimizer

Adam est choisi car :

- Peu gourmand en mémoire
- Tuning des hyperparamètres simplifié

L'hyperparamètre learning\_rate est à 0.0001

### 3. Résultats

#### 3.1 Accuracy sur le jeu de tests de train et validation

#### 3.2 Accuracy sur le jeu de tests après entraînement

#### 3.3 Matrice de confusion et classification report



# 5) Analyse des images

## Comparatif

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-
EfficientNetB0	29 MB	-	-	5,330,571	-
EfficientNetB1	31 MB	-	-	7,856,239	-
EfficientNetB2	36 MB	-	-	9,177,569	-
EfficientNetB3	48 MB	-	-	12,320,535	-
EfficientNetB4	75 MB	-	-	19,466,823	-
EfficientNetB5	118 MB	-	-	30,562,527	-
EfficientNetB6	166 MB	-	-	43,265,143	-
EfficientNetB7	256 MB	-	-	66,658,687	-

Très bon positionnement taille vs accuracy

Très bon classement selon  
<https://paperswithcode.com/paper/self-training-with-noisy-student-improves>



1) Présentation

2) Analyse SIFT



3) Analyse CNN



## 5) Analyse CNN

CNNTransfer Learning Xception: StandardDataSet No Data Augmentation

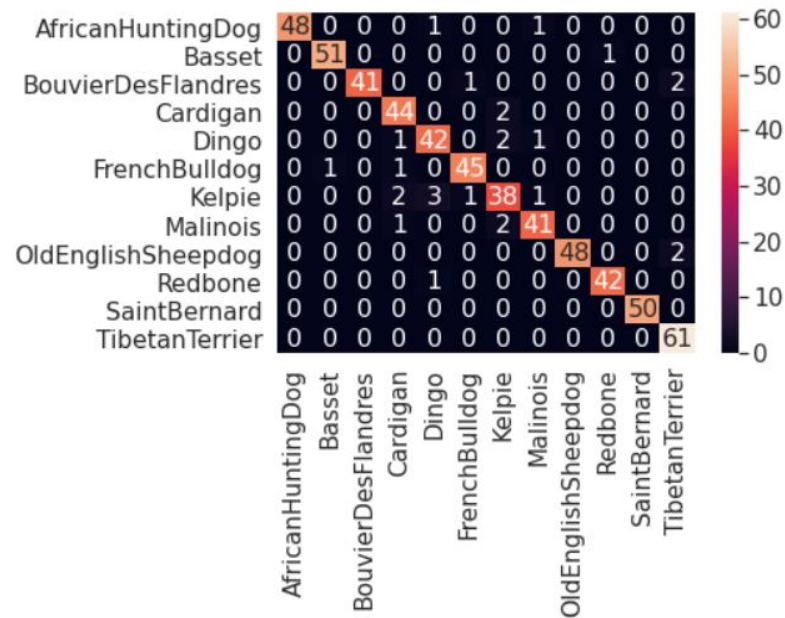


1) Presentation

2) Analyse SIFT



3) Analyse CNN



	precision	recall	f1-score	support
AfricanHuntingDog	1.00	0.96	0.98	50
Basset	0.98	0.98	0.98	52
BouvierDesFlandres	1.00	0.93	0.96	44
Cardigan	0.90	0.96	0.93	46
Dingo	0.89	0.91	0.90	46
FrenchBulldog	0.96	0.96	0.96	47
Kelpie	0.86	0.84	0.85	45
Malinois	0.93	0.93	0.93	44
OldEnglishSheepdog	1.00	0.96	0.98	50
Redbone	0.98	0.98	0.98	43
SaintBernard	1.00	1.00	1.00	50
TibetanTerrier	0.94	1.00	0.97	61
accuracy			0.95	578
macro avg	0.95	0.95	0.95	578
weighted avg	0.95	0.95	0.95	578





## 5) Analyse CNN

CNNTransfer Learning Xception: StandardDataSet With Data Augmentation

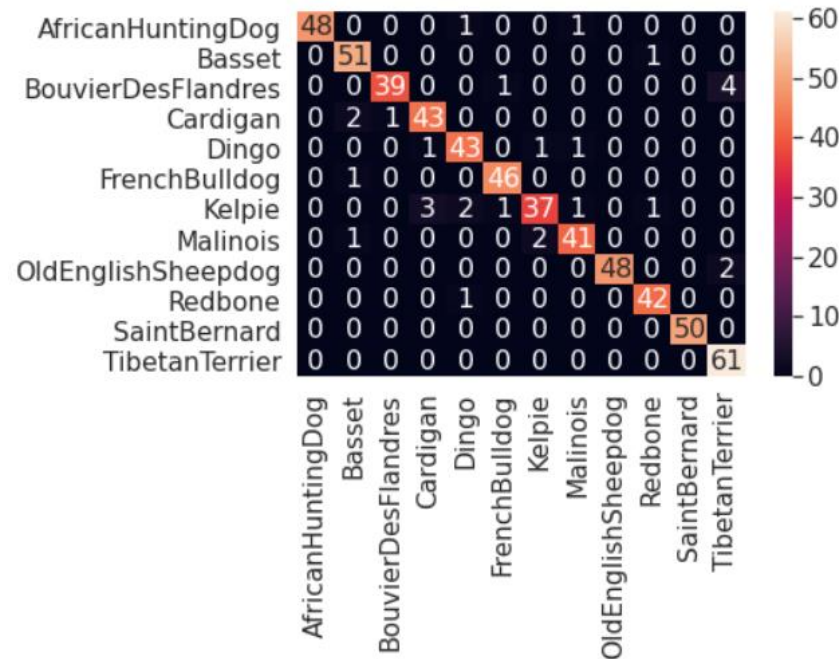


1) Presentation

2) Analyse SIFT



3) Analyse CNN



	precision	recall	f1-score	support
AfricanHuntingDog	1.00	0.96	0.98	50
Basset	0.93	0.98	0.95	52
BouvierDesFlandres	0.97	0.89	0.93	44
Cardigan	0.91	0.93	0.92	46
Dingo	0.91	0.93	0.92	46
FrenchBulldog	0.96	0.98	0.97	47
Kelpie	0.93	0.82	0.87	45
Malinois	0.93	0.93	0.93	44
OldEnglishSheepdog	1.00	0.96	0.98	50
Redbone	0.95	0.98	0.97	43
SaintBernard	1.00	1.00	1.00	50
TibetanTerrier	0.91	1.00	0.95	61
accuracy			0.95	578
macro avg	0.95	0.95	0.95	578
weighted avg	0.95	0.95	0.95	578





# 5) Analyse CNN

CNNTransfer Learning Xception : FullDataSet



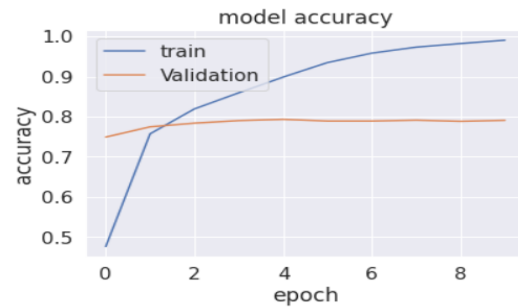
1) Presentation

2) Analyse SIFT



3) Analyse CNN

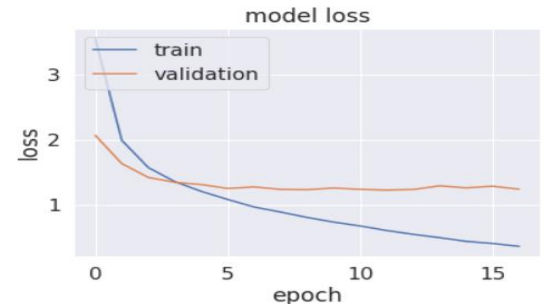
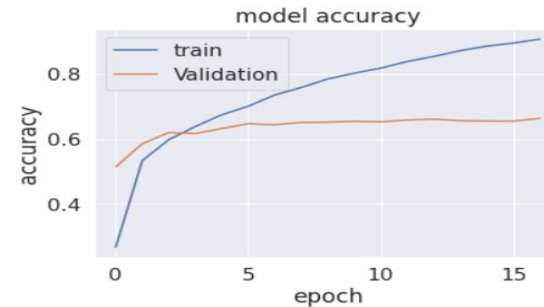
## 1) No Data Augmentation



Time:760.45620984334s

accuracy			0.79	6063
macro avg	0.80	0.79	0.79	6063
weighted avg	0.80	0.79	0.79	6063

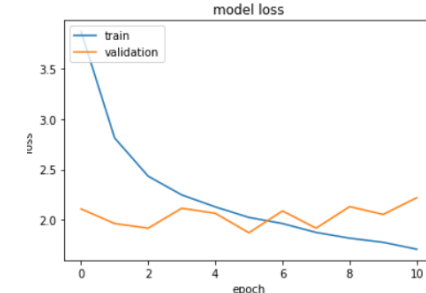
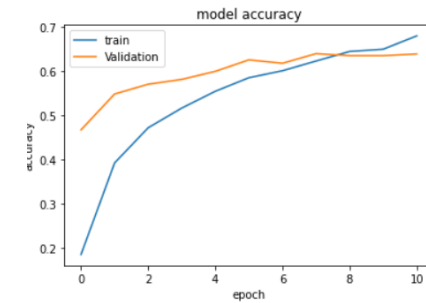
## 2) Data Augmentation



Time:3501.90120927334s

accuracy			0.78	6063
macro avg	0.79	0.78	0.78	6063
weighted avg	0.79	0.78	0.78	6063

## 3) Mixed Up



Time:4406.90290927887s

accuracy			0.75	6063
macro avg	0.75	0.74	0.74	6063
weighted avg	0.75	0.75	0.74	6063



# 5) Analyse CNN

## CNNTransfer Learning Xception Classification Report



1) Presentation

2) Analyse SIFT



3) Analyse CNN

	precision	recall	f1-score	support
Affenpinscher	0.83	0.91	0.87	44
AfghanHound	0.89	0.93	0.91	71
AfricanHuntingDog	0.94	0.98	0.96	50
Airedale	0.80	0.82	0.81	60
AmericanStaffordshireTerrier	0.61	0.75	0.67	48
Appenzeller	0.66	0.57	0.61	44
AustralianTerrier	0.70	0.79	0.74	58
Basenji	0.91	0.82	0.86	62
Basset	0.89	0.75	0.81	52
Beagle	0.75	0.81	0.78	58
BedlingtonTerrier	0.94	0.94	0.94	54
BerneseMountainDog	0.91	0.94	0.92	64
Black	0.84	0.79	0.81	47
BlenheimSpaniel	0.98	0.84	0.90	55
Bloodhound	0.79	0.84	0.81	55
Bluetick	0.83	0.78	0.80	50
BorderCollie	0.72	0.82	0.77	44
BorderTerrier	0.86	0.82	0.84	51
Borzoi	0.86	0.86	0.86	44
BostonBull	0.77	0.93	0.84	54
BouvierDesFlandres	0.72	0.70	0.71	44
Boxer	0.88	0.50	0.64	44
BrabanconGriffon	0.83	0.78	0.80	45
Briard	0.71	0.82	0.76	45
BrittanySpaniel	0.80	0.87	0.83	45
BullMastiff	0.85	0.85	0.85	46
Cairn	0.90	0.81	0.85	58
Cardigan	0.80	0.76	0.78	46
ChesapeakeBayRetriever	0.57	0.86	0.68	49
Chihuahua	0.74	0.71	0.73	45
Chow	0.95	0.91	0.93	58
Clumber	0.91	0.95	0.93	44
CockerSpaniel	0.82	0.77	0.79	47
Collie	0.55	0.62	0.58	45
Curly	0.70	0.86	0.78	44
DandieDinmont	0.92	0.89	0.90	53
Dhole	0.97	0.84	0.90	44
Dingo	0.80	0.85	0.82	46
Doberman	0.84	0.86	0.85	44
EnglishFoxhound	0.72	0.63	0.67	46
EnglishSetter	0.92	0.74	0.82	47
EnglishSpringer	0.84	0.91	0.88	47
Entlebucher	0.73	0.72	0.72	60
EskimoDog	0.50	0.32	0.39	44
Flat	0.80	0.78	0.79	45
FrenchBulldog	0.87	0.85	0.86	47
GermanShepherd	0.85	0.78	0.81	45
GermanShort	0.80	0.89	0.84	45
GiantSchnauzer	0.75	0.65	0.70	46
GoldenRetriever	0.77	0.75	0.76	44
GordonSetter	0.87	0.89	0.88	45
GreatDane	0.82	0.80	0.81	46
GreatPyrenees	0.65	0.76	0.70	63
GreaterSwissMountainDog	0.73	0.76	0.74	49
Groenendael	0.82	0.84	0.83	44
IbizanHound	0.90	0.84	0.87	55
IrishSetter	0.96	0.93	0.95	46
IrishTerrier	0.73	0.86	0.79	50
IrishWaterSpaniel	0.84	0.84	0.84	44
IrishWolfhound	0.76	0.80	0.78	64
ItalianGreyhound	0.69	0.74	0.71	54
JapaneseSpaniel	0.84	0.94	0.89	54
Keeshond	0.96	0.96	0.96	46
Kelpie	0.66	0.69	0.67	45
KerryBlueTerrier	0.89	0.89	0.89	53
Komondor	0.97	0.82	0.89	45
Kuvasz	0.84	0.73	0.78	44
LabradorRetriever	0.71	0.68	0.69	50
LakelandTerrier	0.83	0.67	0.74	58
Leonberg	0.92	0.90	0.91	62
Lhasa	0.62	0.55	0.58	55
Malamute	0.55	0.83	0.66	52
Malinois	0.81	0.77	0.79	44
MalteseDog	0.88	0.87	0.87	75
MexicanHairless	0.86	0.96	0.91	46
MiniaturePinscher	0.77	0.85	0.81	54
MiniaturePoodle	0.68	0.37	0.48	46
MiniatureSchnauzer	0.70	0.82	0.76	45
Newfoundland	0.68	0.74	0.71	58
NorfolkTerrier	0.78	0.57	0.66	51
NorwegianElkhound	0.80	0.90	0.85	58
NorwichTerrier	0.73	0.65	0.69	54
OldEnglishSheepdog	0.87	0.94	0.90	50
Otterhound	0.91	0.73	0.81	44
Papillon	0.96	0.76	0.85	58
Pekinese	0.77	0.77	0.77	44
Pembroke	0.77	0.89	0.82	53
Pomeranian	0.91	0.92	0.92	65
Pug	0.84	0.88	0.86	59
Redbone	0.67	0.65	0.66	43
RhodesianRidgeback	0.62	0.71	0.66	51
Rottweiler	0.86	0.93	0.89	45
SaintBernard	0.94	0.98	0.96	50
Saluki	0.77	0.95	0.85	59
Samoyed	0.93	0.97	0.95	64
Schipperke	0.77	0.91	0.84	45
ScotchTerrier	0.93	0.83	0.87	46
ScottishDeerhound	0.86	0.83	0.84	69
SealyhamTerrier	0.91	0.88	0.90	60
ShetlandSheepdog	0.86	0.67	0.76	46
Shih	0.67	0.71	0.69	63
SiberianHusky	0.55	0.42	0.48	57
SilkyTerrier	0.79	0.69	0.73	54
Soft	0.73	0.70	0.71	46
StaffordshireBullterrier	0.61	0.54	0.57	46
StandardPoodle	0.71	0.72	0.72	47
StandardSchnauzer	0.59	0.76	0.67	46
SussexSpaniel	0.98	0.91	0.94	44
TibetanMastiff	0.89	0.87	0.88	45
TibetanTerrier	0.80	0.67	0.73	61
ToyPoodle	0.64	0.66	0.65	44
ToyTerrier	0.88	0.59	0.71	51
Vizsla	0.77	0.82	0.80	45
WalkerHound	0.65	0.58	0.61	45
Weimaraner	0.82	0.89	0.86	47
WelshSpringerSpaniel	0.80	0.89	0.84	44
WestHighlandWhiteTerrier	0.76	0.94	0.84	50
Whippet	0.70	0.64	0.67	55
Wire	0.86	0.80	0.83	46
YorkshireTerrier	0.65	0.75	0.70	48
accuracy			0.79	6063
macro avg	0.80	0.79	0.79	6063
weighted avg	0.80	0.79	0.79	6063



## 5) Analyse CNN

CNNTransfer Learning efficientnetb7 StandardDataSet No Data Augmentation

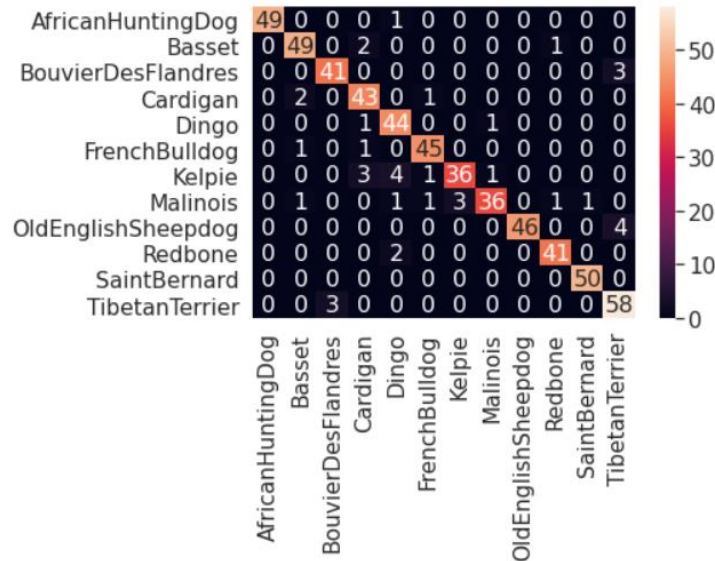


1) Présentation

2) Analyse SIFT



3) Analyse CNN

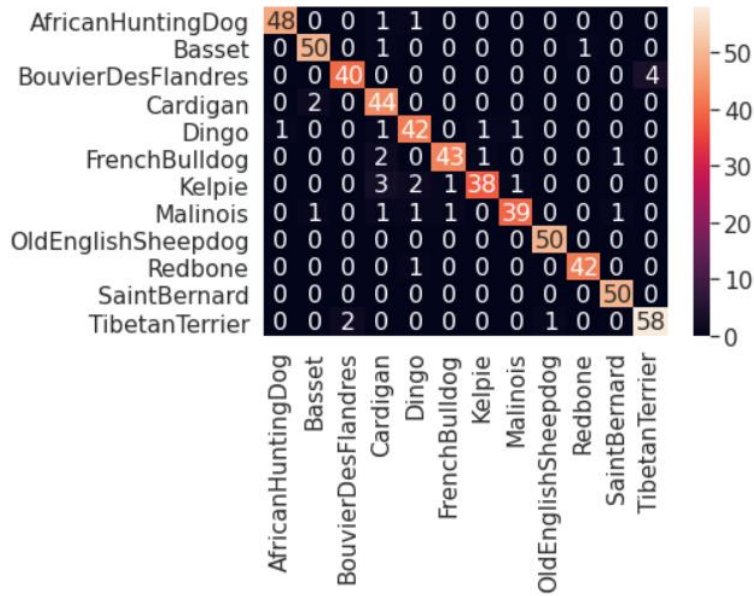
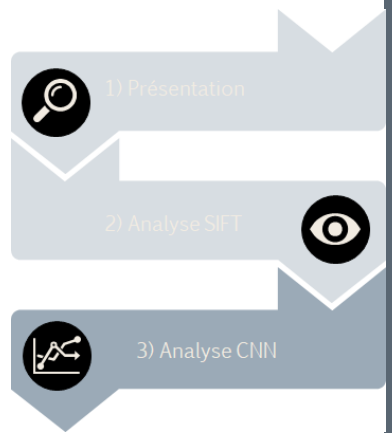


	precision	recall	f1-score	support
AfricanHuntingDog	1.00	0.98	0.99	50
Basset	0.92	0.94	0.93	52
BouvierDesFlandres	0.93	0.93	0.93	44
Cardigan	0.86	0.93	0.90	46
Dingo	0.85	0.96	0.90	46
FrenchBulldog	0.94	0.96	0.95	47
Kelpie	0.92	0.80	0.86	45
Malinois	0.95	0.82	0.88	44
OldEnglishSheepdog	1.00	0.92	0.96	50
Redbone	0.95	0.95	0.95	43
SaintBernard	0.98	1.00	0.99	50
TibetanTerrier	0.89	0.95	0.92	61
accuracy			0.93	578
macro avg	0.93	0.93	0.93	578
weighted avg	0.93	0.93	0.93	578



## 5) Analyse CNN

CNNTransfer Learning efficientnetb7 StandardDataSet With Data Augmentation



	precision	recall	f1-score	support
AfricanHuntingDog	0.98	0.96	0.97	50
Basset	0.94	0.96	0.95	52
BouvierDesFlandres	0.95	0.91	0.93	44
Cardigan	0.83	0.96	0.89	46
Dingo	0.89	0.91	0.90	46
FrenchBulldog	0.96	0.91	0.93	47
Kelpie	0.95	0.84	0.89	45
Malinois	0.95	0.89	0.92	44
OldEnglishSheepdog	0.98	1.00	0.99	50
Redbone	0.98	0.98	0.98	43
SaintBernard	0.96	1.00	0.98	50
TibetanTerrier	0.94	0.95	0.94	61
accuracy			0.94	578
macro avg	0.94	0.94	0.94	578
weighted avg	0.94	0.94	0.94	578



# 5) Analyse CNN

CNNTransfer Learning efficientnetb7 : FullDataSet



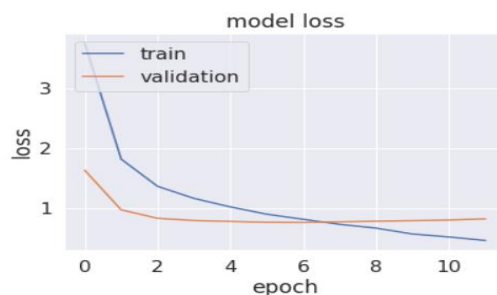
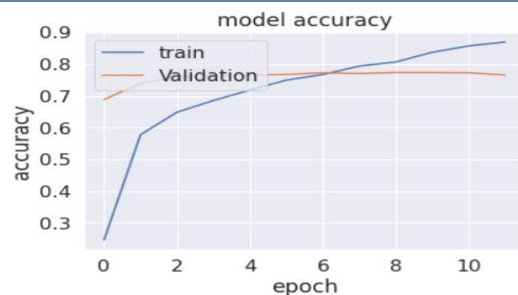
1) Présentation

2) Analyse SIFT



3) Analyse CNN

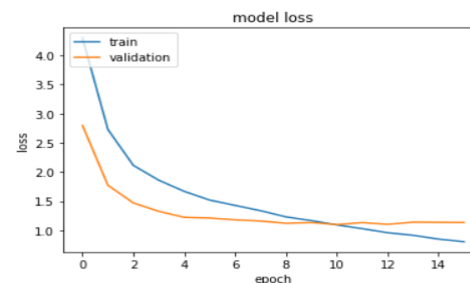
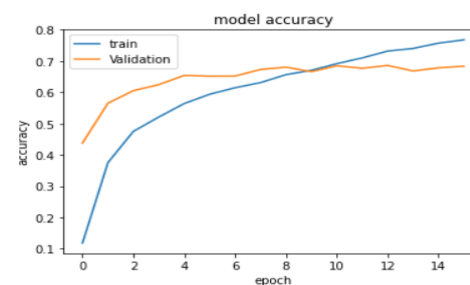
## 1) No Data Augmentation



Time: 1531.5925679206848s

accuracy			0.78	6063
macro avg	0.78	0.77	0.77	6063
weighted avg	0.79	0.78	0.78	6063

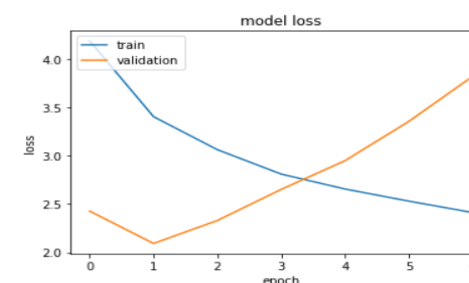
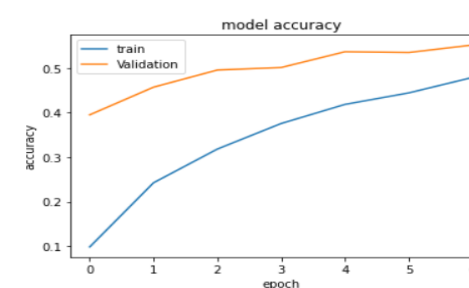
## 2) Data Augmentation



Time : 4071.7993955612183s

accuracy			0.77	6063
macro avg	0.78	0.76	0.76	6063
weighted avg	0.78	0.77	0.77	6063

## 3) Mixed Up



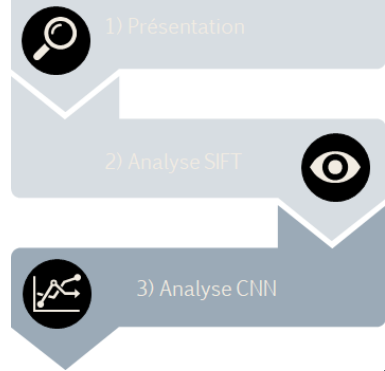
Time : 3915.4189338684087s

accuracy			0.56	6063
macro avg	0.62	0.55	0.53	6063
weighted avg	0.61	0.56	0.53	6063



# 5) Analyse CNN

Comparatif



	Xception			EfficientB7		
	NoDataAugmentation	DataAugmentation	MixedUp	NoDataAugmentation	DataAugmentation	MixedUp
Accuracy	0,80	0,79	0,75	0,78	0,78	0,62
Time (s)	760	3501	4406	1531	4071	3915





## 5) Analyse CNN

Exemples d'erreurs detectées pour le modèle le plus performant



1) Présentation

2) Analyse SIFT



3) Analyse CNN



Catégorie réelle	Malamute
Catégorie détectée	Siberian Husky



Catégorie réelle	American Staffordshireterrier
Catégorie détectée	Staffordshire Bullterrier



Catégorie réelle	Shetland SheepDog
Catégorie détectée	Collie



Catégorie réelle	Siberian Husky
Catégorie détectée	Malamute



Catégorie réelle	Staffordshire Bullterrier
Catégorie détectée	American Staffordshireterrier



Catégorie réelle	Collie
Catégorie détectée	Shetland SheepDog



## 5) Conclusion

### Modèles

- Les modèles de type CNN sont très performants.
- Le transfer learning permet d'avoir des résultats très bons par une simple configuration.

### Pré-traitements

- Des pré-traitements tels que Rotation, Zoom, Std, .... ont leur intérêt sur des modèles de type SIFT
- Pour des modèles de type CNN, ces pré-traitements ne sont pas toujours nécessaires (dependence du jeu de données).





## 7) Prochaines étapes

### **1** *Revue EfficientNetB7*

Revoir la configuration du modèle EfficientNetB7 afin d'avoir de meilleurs résultats (une accuracy d'environ 90% est possible)

### **2** *Images de l'association*

Application du meilleur modèle sur les images de l'association. Une validation manuelle est faite pour s'assurer des bonnes races.

### **3** *Stabilisation*

Alimentation des Nouvelles images au modèle

Validation de la race proposée par un membre de l'association



## 8) Environnements techniques



kaggle