



Parcours Data Scientist

Projet 7 : Classification de races de chiens



Sommaire

- Présentation et Objectifs
- Exploration/Préparation
- Modèles Classiques
- Réseaux de Neurones
- Kaggle
- API
- Pistes d'évolutions
- Conclusion



Présentation et Objectifs

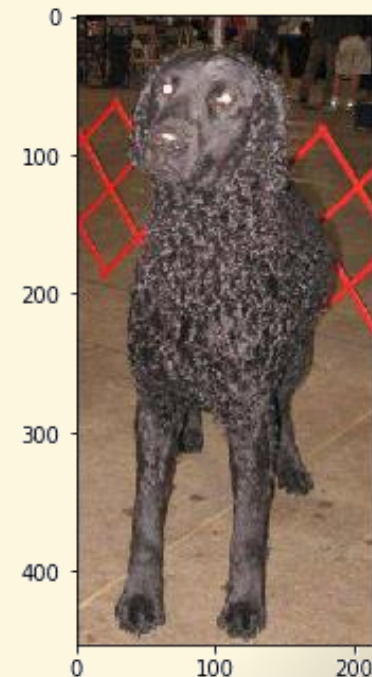
- Objectifs
 - Prédiction de la Race d'un chien sur une image
 - Méthode Classique
 - Réseaux de Neurones
 - Mise en place d'une API
- Présentation
 - Basé sur un dataset labélisé de photos de chiens
 - Traitement d'images

Exploration/Préparation

- Dataset
 - 10 222 images (Kaggle : 10357 images de test)
 - 120 races
 - Balance correcte mais pas optimale
 - 66 -> 126 images / race
 - Dimensions
 - 120x102 -> 3264x2448
 - Portrait et Landscape
 - Ratio variable
 - <0,5 jusqu'à >2

Exploration/Préparation

- Préparation
 - Redimensionnement manuel
 - Si >2 ou $<0,5$



Exploration/Préparation

- Préparation
 - Data Augmentation
 - Flip Vertical
 - Supprimé (overfitting sur Pre-trained Network)



Modèle Classique

- Concaténation de 4 types de données
 - Détection de features
 - Attribut Couleur
 - Attribut Texture
 - Moments



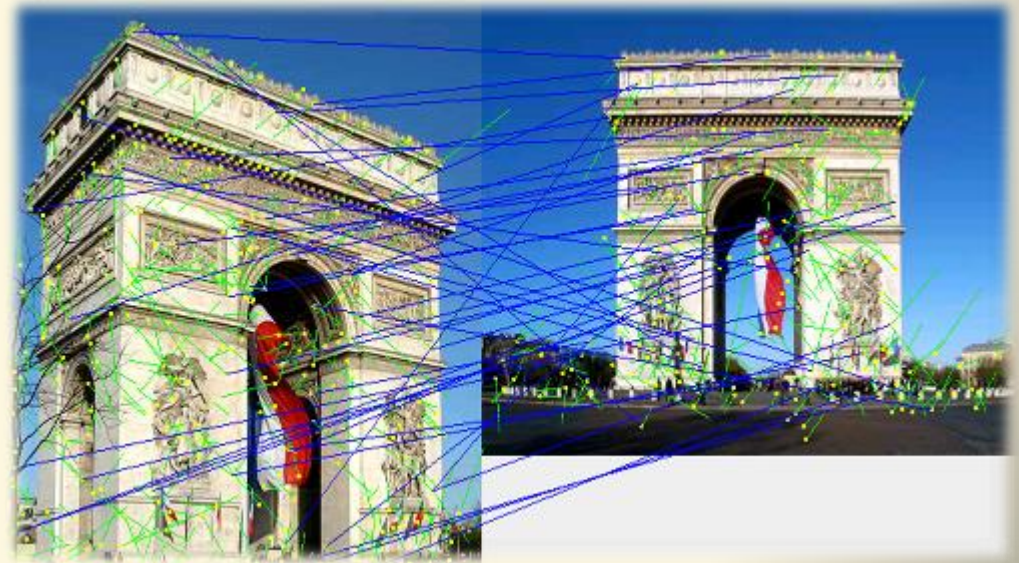
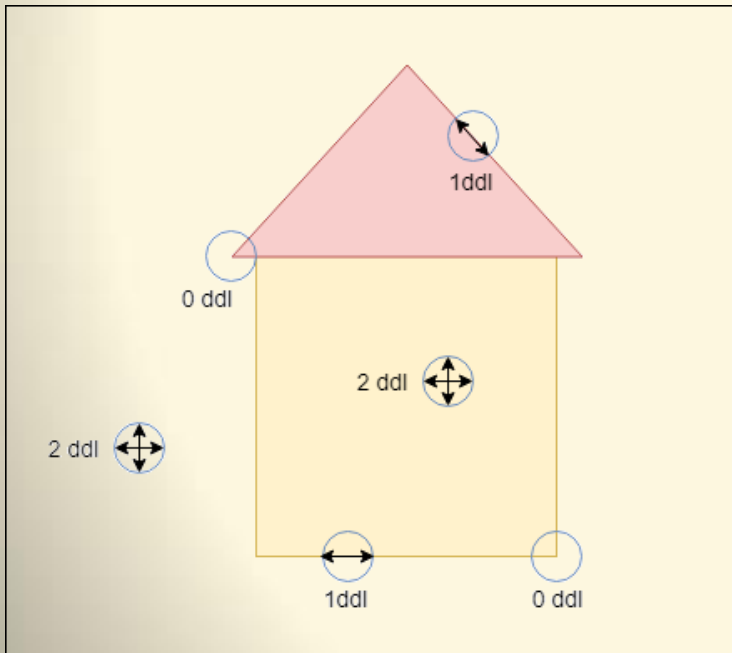
Modèle Classique

- Feature SIFT + Visual BoW
 - SIFT keypoints
 - Clustering
 - BoW



Modèle Classique

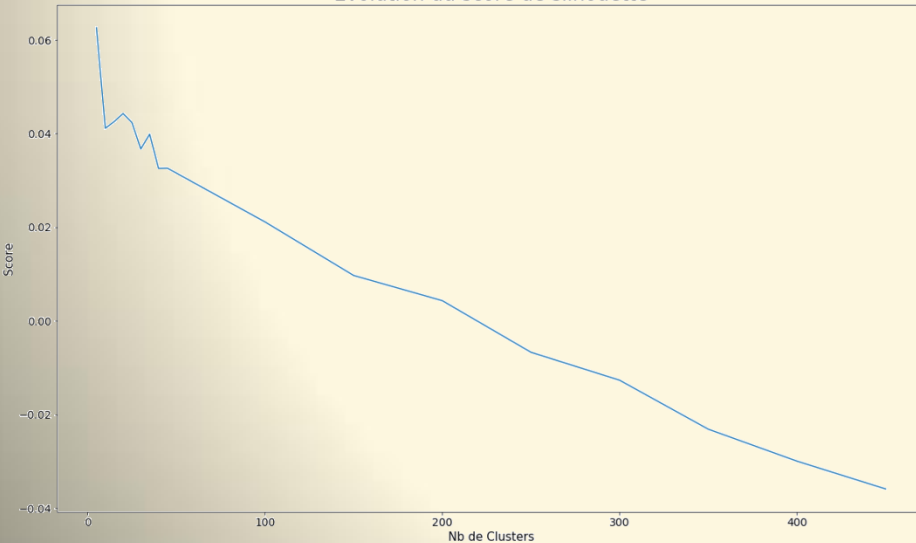
- SIFT
 - Extraire les points facilement « détectables »



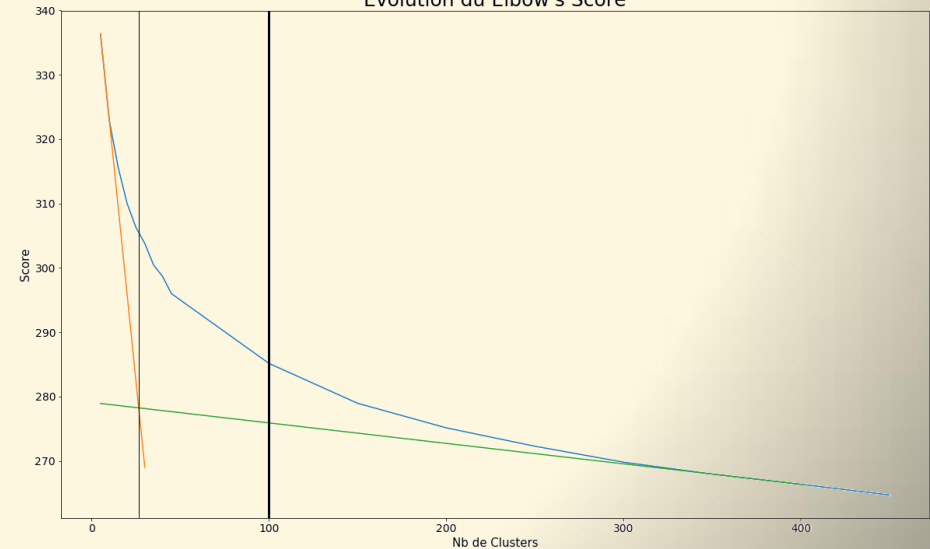
Modèle Classique

- Clustering (pas de scaling)
 - Env. 1k features / images
 - Sélection : 100 Clusters
 - Optimal : 27 Clusters

Evolution du score de silhouette

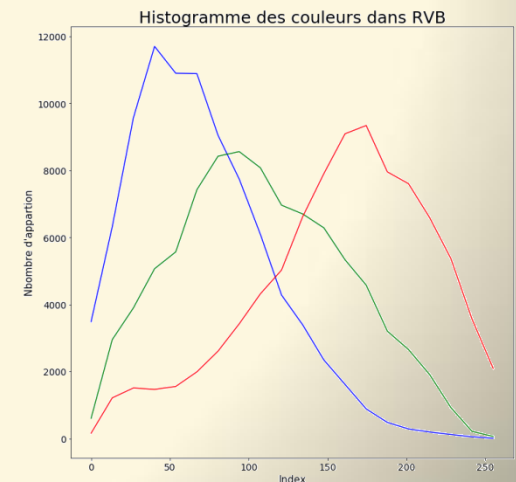
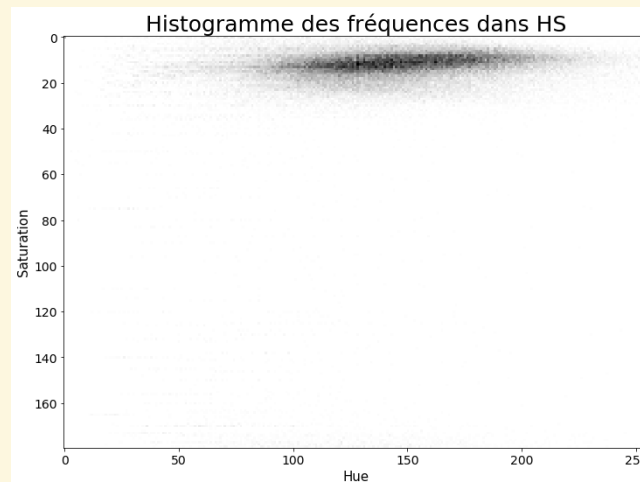


Evolution du Elbow's Score



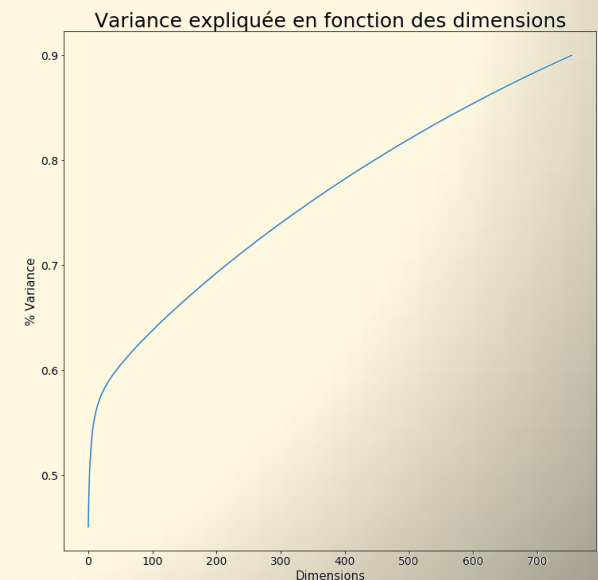
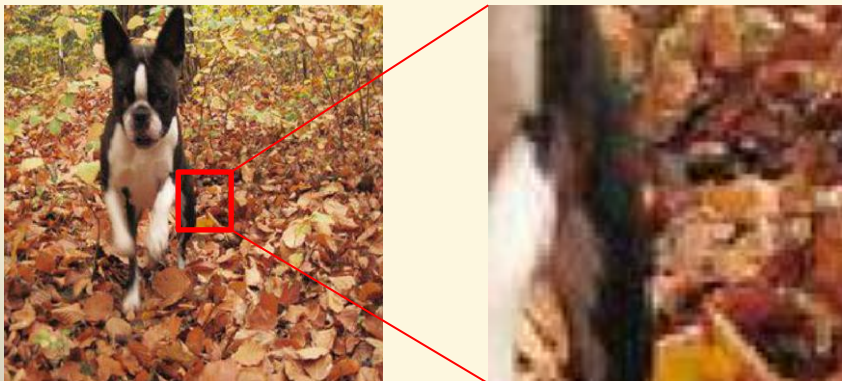
Modèle Classique

- Attribut Couleur
 - Conversion RGB à HSV
 - Histogramme Hue/Saturation
 - 180 x 256 valeurs
 - Réduction PCA : 90% variance = 151 dimensions



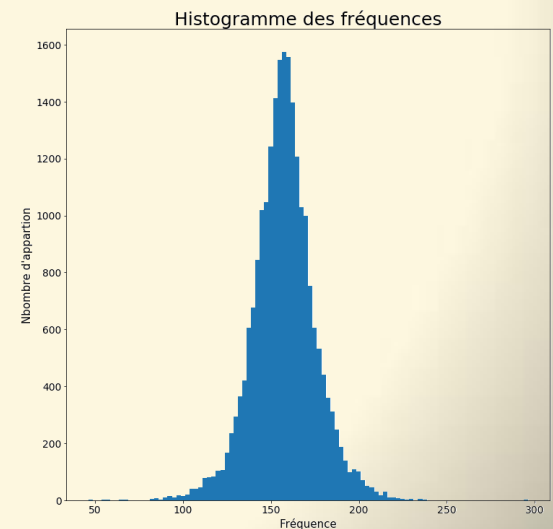
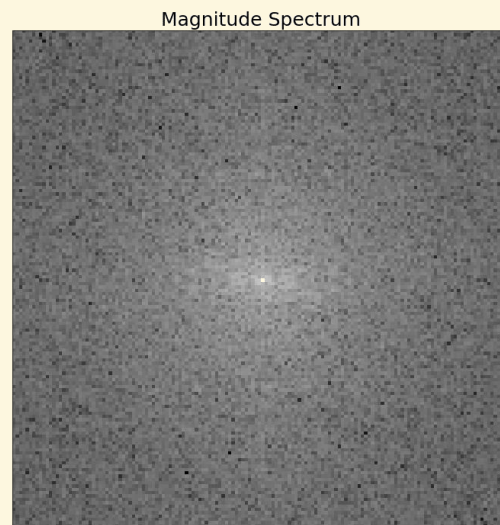
Modèle Classique

- Attribut Texture
 - Analyse FFT
 - 1 valeur par pixel => 90k valeurs
 - Tentative 1 : réduction PCA
 - Variance 90 % = 4500 dimensions
 - Tentative 2 : Sélection patch au milieu + PCA
 - Variance 63% = 100 dimensions



Modèle Classique

- Attribut Texture
 - Histogramme FFT
 - 100 bins = 100 dimensions



Modèle Classique

- Moments
 - Récupération des moments par couleurs
 - $3 * 24$ features additionnelles
 - Très variables (de 10^7 à 10^{-7})
 - Standard Scaling par features
 - 10 Moments Ordre 1, 2, 3
 - 7 Moments Centrés Ordre 1, 2, 3
 - 7 Moments Centrés Normalisés Ordre 1, 2, 3

Modèle Classique

- Evaluation
 - Concaténation des matrices
 - $X = 10222 \times 523 (100+151+100+100+3*24)$
 - $Y = 10222 \times 120 (120 \text{ races})$
 - Dummy Classifieurs (Stratified)
 - Full Dataset :
 - 0,93 % (train)
 - 0,98% (test)
 - Réduction à 5 classes :
 - 20,8% (train)
 - 23,5%(test)

Modèle Classique

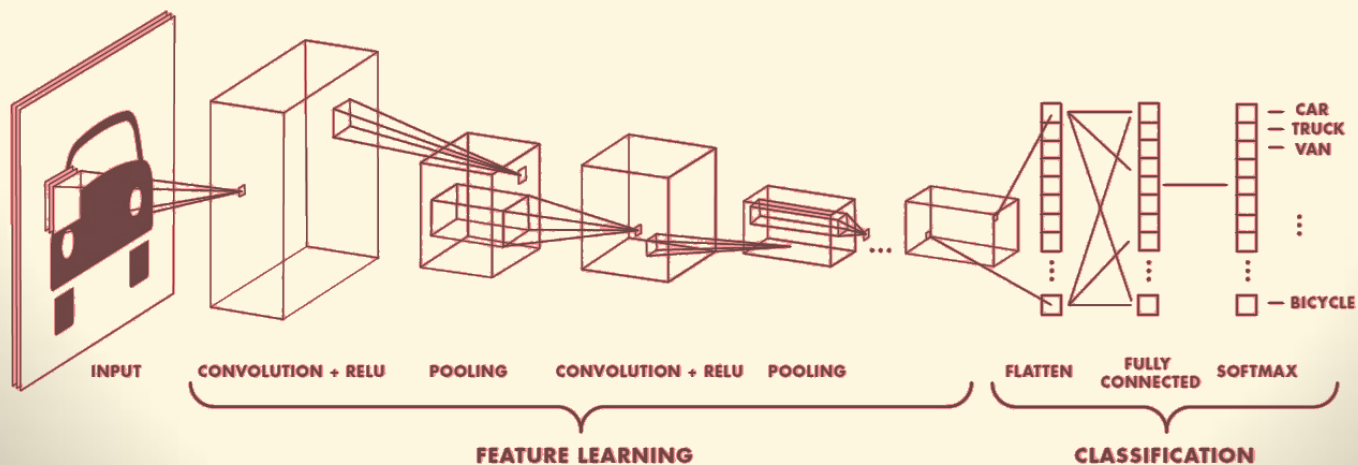
- Résultats
 - Multiples modèles testés
 - Tree, Ensemble, KNN, Modèle Linéaire (SGDC), Modèle Non linéaire (SVC)
 - Très gros overtiffing
 - Régulation
 - réduction du score test set
 - Similaire au Dummy Classifieur avec Reg.
 - Résultats
 - 2 à 3 % sur full dataset (env. 3 x mieux que naif)
 - 35 à 37% sur full dataset (env. 1,7 x mieux que naif)

Modèle Classique

- Modèle 2 :
 - 1 classifieur par type de données
 - HS histogramme + PCA => KNN
 - Visual BoW => MultinomialNB
 - Histogrammes fréquences => KNN custom
 - Swain and Ballard, KL divergence ou chi2
 - Moments => Normalisation et KNN
 - Somme des probabilités + normalisation
 - 2,73 % sur test set Vs 2,61% précédemment

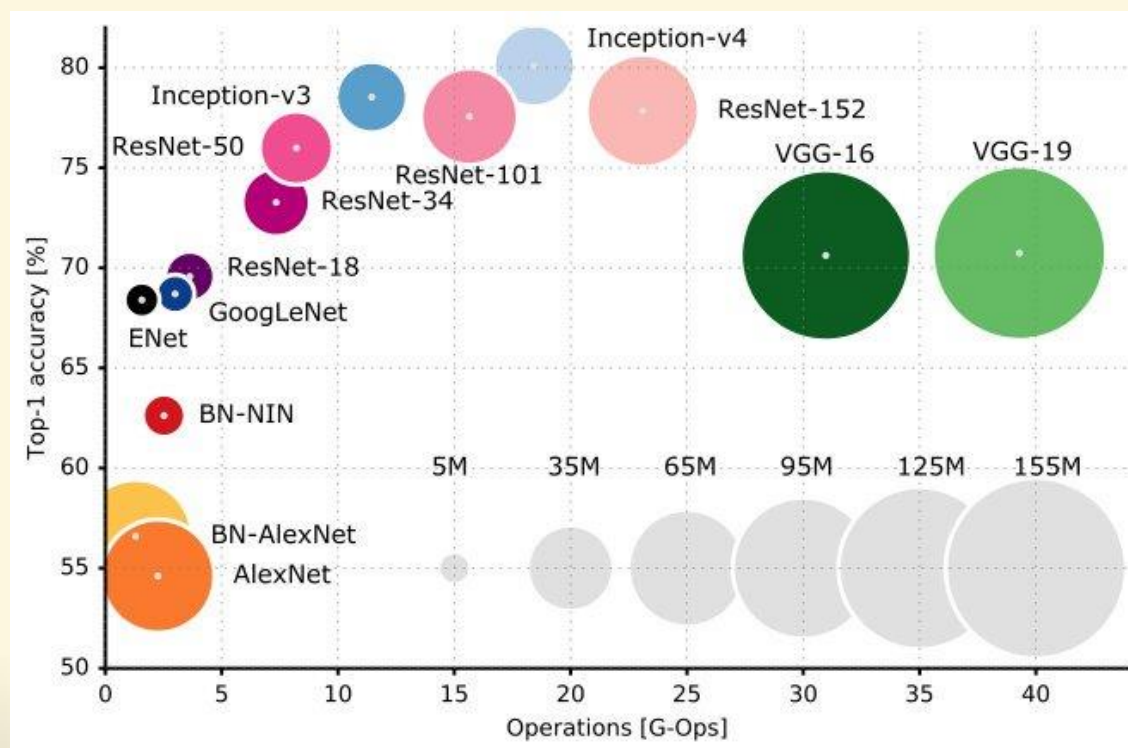
Réseaux de Neurones

- 3 parties
 - Extraction des features
 - Multiples modèles
 - Comparaison à classifieur unique
 - Test de Classifieurs



Réseaux de Neurones

- Extraction des features
 - Pre-trained Models



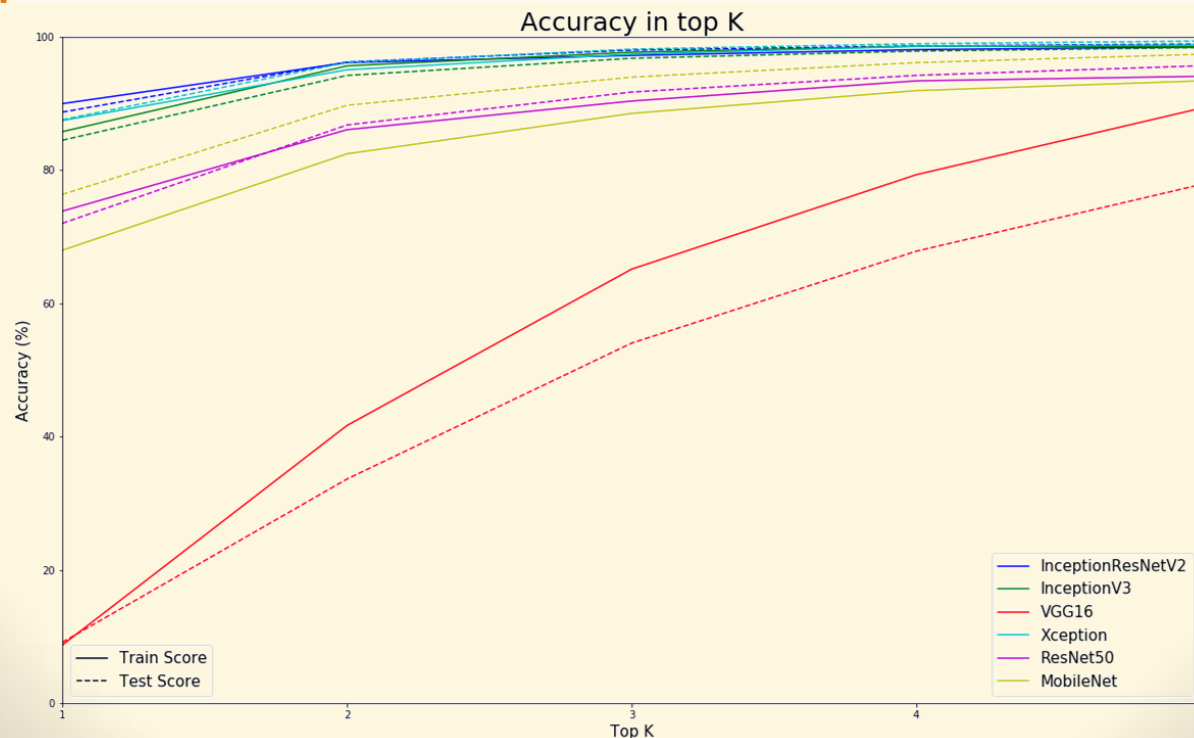
Réseaux de Neurones

- Extraction des features
 - Pre-trained Models

Model	Size	Params	Depth	Input	Output
Xception	88Mo	22,9 M	126*	299x299	2048
VGG16	528Mo	138,3 M	23	224x224	512
ResNet50	99Mo	25,6 M	168*	224x224	2048
InceptionV3	92Mo	23,8 M	159*	299x299	2048
InceptionResNetV2	215Mo	55,8 M	572*	299x299	1536
MobileNet	17Mo	4,2 M	88*	224x224	1024

Réseaux de Neurones

- Comparaison à classifieur unique
 - D512 – D0,5 – D512 – D120+Softmax
 - Top 1 à 5
















Réseaux de Neurones

- Test de Classifieur
 - SGDC : Reg. => Chute du %
 - SVC : Top % - Mauvais Loss
 - NN : D200+swish -D0,5 - D120+softmax

Modèle		Extra Info	Train %	Test %	Loss Train %	Loss Test %
Linéaire	SGCD	Sans Reg.	94,2	90,3	0,479	1,326
Non-Linéaire	SVC	Probability = True	91,6	91,5	0,841	0,88
Ensemble	ExtraTrees RandomForest	Plusieurs Depth	Env. 80%	Env. T - 5%	N/A	N/A
Non-Linéaire	Neural Network	Plusieurs Topologies	91,71	91,20	0.2816	0.2821

Kaggle

- 10357 images sans labels
 - 204 / 847

199	▼ 18	psnjiki		0.25871	15	3mo
200	▼ 18	Kirill Talalaev (TFS)		0.26392	11	2mo
201	▼ 18	EE258_GC_IC	 	0.26461	25	1mo
202	▼ 18	BillJohnson		0.26545	38	1mo
203	▼ 18	CT		0.26811	3	2mo
204	▲ 177	Nicolas MINE		0.26867	7	3m
205	▼ 19	Roman Kornev		0.27036	4	2mo
206	▼ 19	mtcmr2		0.27128	14	2mo
207	▼ 19	PlodHL		0.27295	1	1mo
208	▼ 19	irving		0.27297	4	1mo
209	▼ 19	James Liu		0.27383	7	1mo
210	▼ 19	earth		0.27391	17	1mo

MNIST

- 50k images train + 10k test
- 28x28 pixels
- Model:
 - Conv2D + relu => 32 layers de 26x26
 - 32 filtres – kernel 3 – strides 1 – padding SAME
 - Conv2D+ relu => 64 layers de 24x24
 - 64 filtres – kernel 3 – strides 1 – padding SAME
 - Max Pooling => 64 layers de 14x14 (à cause du padding VALID)
 - Kernel 1-2-2-1 – Strides 1-2-2-1 – Padding VALID
 - Flattening
 - Dense 128 + relu
 - Dense 10 + softmax
- 99.08% sur le test set
- 99.585% sur Kaggle (Classement entre 193 et 207/1943)

API

- Modèle CNN
- 1 à N images
 - Pre-process 1 par 1
 - Merge (n, 299, 299, 3)
 - Extraction features (n, 1536)
 - Prediction (n, 120)
 - Affichage par image du top 5 avec %
- `python classifieur.py img1 img2 ... imgN`

API



```
../test/0a4ef19459cd2100977b052de5f46231.jpg  
Rank 1 - silky_terrier (97.950%)  
Rank 2 - australian_terrier (1.388%)  
Rank 3 - yorkshire_terrier (0.529%)  
Rank 4 - greater_swiss_mountain_dog (0.015%)  
Rank 5 - norfolk_terrier (0.011%)
```

```
../test/2f09a0cc0902a43ba8a410c259fb4309.jpg  
Rank 1 - chihuahua (98.425%)  
Rank 2 - papillon (0.715%)  
Rank 3 - toy_terrier (0.543%)  
Rank 4 - pomeranian (0.028%)  
Rank 5 - mexican_hairless (0.025%)
```



Pistes d'évolutions

- Modèle Classique
 - Fréquences sur image non réduites
 - Autre descripteurs au lieu de SIFT (SURF / ORB)
 - Contours ?
 - Différentes focales

Pistes d'évolutions

- CNN
 - Nettoyage images : Regression (YOLO)
 - Risque mauvais dimensions
 - Modèle manuel pas fait
 - Trop peu de données
 - Vanishing Gradient
 - Beaucoup d'images nécessaires (imagenet)
 - Entraînement du pre-trained
 - Modifier layers de l'extracteur
 - Très lent : 1j =Env. 50 Epochs -> 64% Acc

Pistes d'évolutions

- Analyse de l'overfitting
 - Class Activation Mapping
 - Possible sur VGG et MobileNet
 - Pas de classifieur Customisé
 - Remonte le réseau pour l'analyse

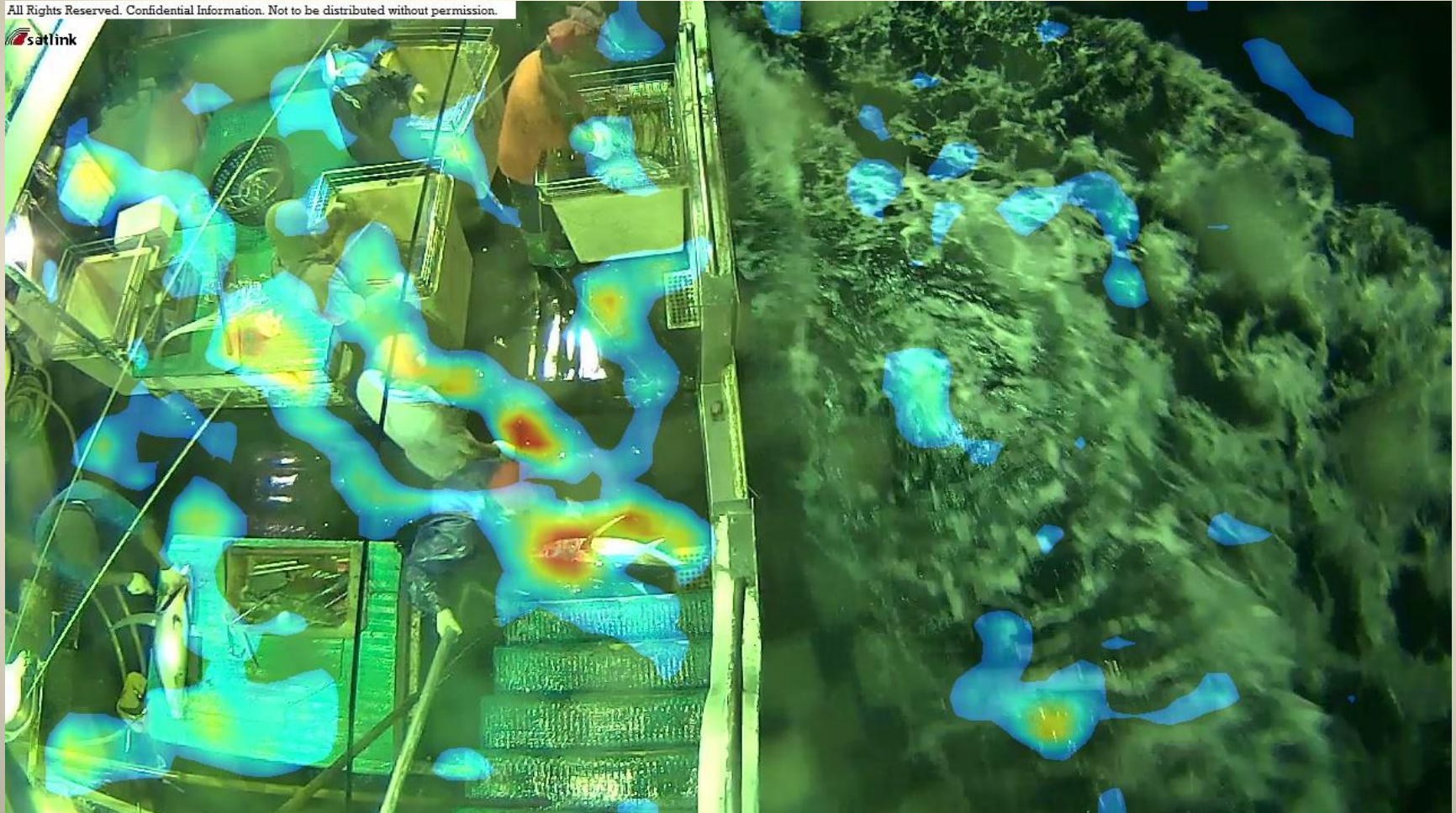
Grad-CAM for "Cat"



Grad-CAM for "Dog"



Pistes d'évolutions



Conclusion

- Dataset complexe (classification pointue)
 - Uniquement des chiens
 - Beaucoup de races vs taille du dataset
- Modèle classique très peu performant
- CNN très performants (Transfer Learning)
- Performances réduite « en partant de zéro »
- Pas encore de moyen de contrôle sur les CNN
 - Hormis CAM sur quelques modèles précis

