

# **Soutenance**

# **Reconnaissance Images**

# **& Vidéos**

---

Application de reconnaissance automatique d'icônes

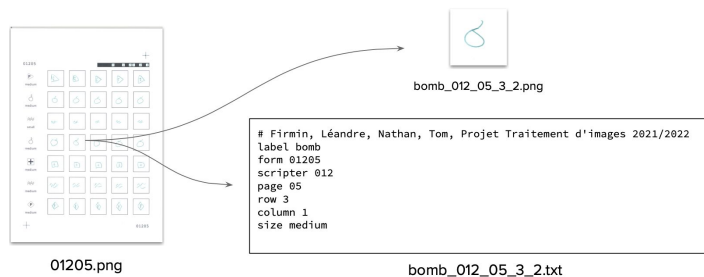
Léandre Le Bizec & Nathan Moureaux

16/02/2023

# Un projet en 2 parties

S7

Traitement de l'image

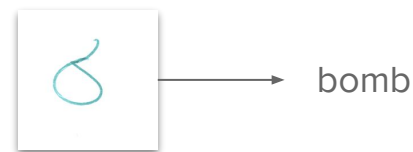


Génération d'une base vérité terrain

=> 26 943 imagerettes

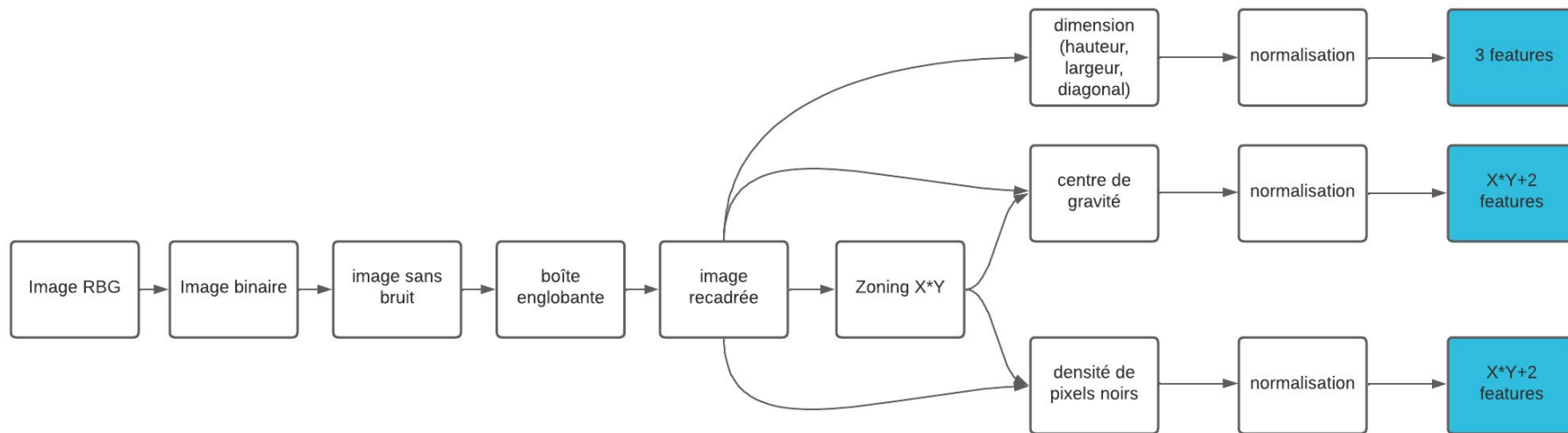
S8

Reconnaissance de l'image



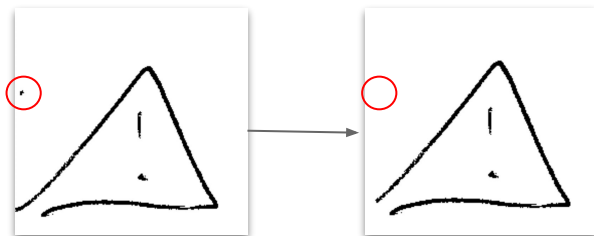
Affectation d'une image à une classe

# Extraction des features - La chaîne



# Extraction des features - Méthodes

Suppression du bruit



pixel isolé -> flou gaussien  
groupe de pixel -> distance moyenne au centre de gravité

Bounding Box



Zoning

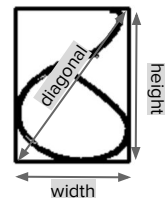


# Extraction des features - Données retenues

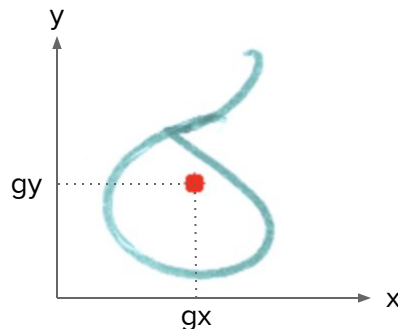
Densité

$$\frac{\text{Nb pixels noirs}}{\text{Nb pixels total}}$$

Dimension



Coordonnées du centre de gravité



# Algorithmes utilisés

- kNN => pour rechercher des meilleurs features
- MLP
- RandomForest
- SVM

=> Tous les classifieurs sont testés avec une validation croisée de 10 sous-échantillons

# kNN (k Nearest Neighbors)

Recherche de la meilleure dimension pour le zoning en se basant sur l'algorithme des kNN sur weka

- Avec  $x=y$ :

Dimension	1x1	2x2	3x3	4x4	5x5	6x6	7x7
Pourcentage d'image bien classifiée	50.464 %	84.0497 %	94.3987 %	95.4491 %	95.5457 %	95.9243 %	95.7164 %
Nb de features	9	18	33	54	81	114	153

- Avec  $x \neq y$ :

Dimension	2x3	3x2	3x4	4x3	4x5	5x4	3x5
Pourcentage d'image bien classifiée	90.3871 %	90.5467 %	94.429 %	94.7667 %	95.1342 %	95.2344 %	94.544 %
Nb de features	24	24	42	42	66	66	51

- Avec des k différents :

k	1	2	3	4	5	6
Précision (%)	95.4491	94.6845	95.0711	94.9152	94.9375	94.7816

# kNN (k Nearest Neighbors)

Recherche du meilleure zoning avec sélection d'attributs en se basant sur l'algorithme des kNN  
=> RankedFeatures avec Weka

Dimension	4*4		5*5	
Nombre de features	54	30	81	54
Précision (%)	95,4491 %	92,4154 %	95,5457 %	94,5784 %



# MLP (Multi Layer Perceptron)

Recherche de la meilleure paramètre pour le classifieur MLP avec notre base de donné

- learning rate

learning rate	0.05	0.1	0.2	0.5
Précision (%)	89.8456 %	93.8382 %	91.2548	78.1514 %

- temps d'apprentissage

temps d'apprentissage	200 s	250 s	300 s	500 s
Précision (%)	93.7565 %	93.8382 %	93.7899 %	93.5969 %

- Couche

Couche	2(20)	5(5)	6(6)	6(8)	8(8)
Précision (%)	92.2383 %	91.4784 %	93.0561 %	93.8382%	92.4536 %

# RandomForest & SVM (Support Vector Machine)

- RandomForest :

maxDepth	$\infty$	10	15	20	30
Précision (%)	95.0891	92.7988	95.0594	95.193	95.0891

- SVM :

Avec les paramètres optimaux  
déterminés par le gridSearch



97.0964 %

# Résultats avec la base de tests

Algorithme	kNN	RandomForest	MLP	SVM
Paramètres	distance : euclidienne k = 1	maxDepth = 20	learning rate = 0.1 couche = 6*8 temps = 250 s	gridSearch
Précisions	92.5714 %	87.1429 %	94.2857 %	82.7751 %

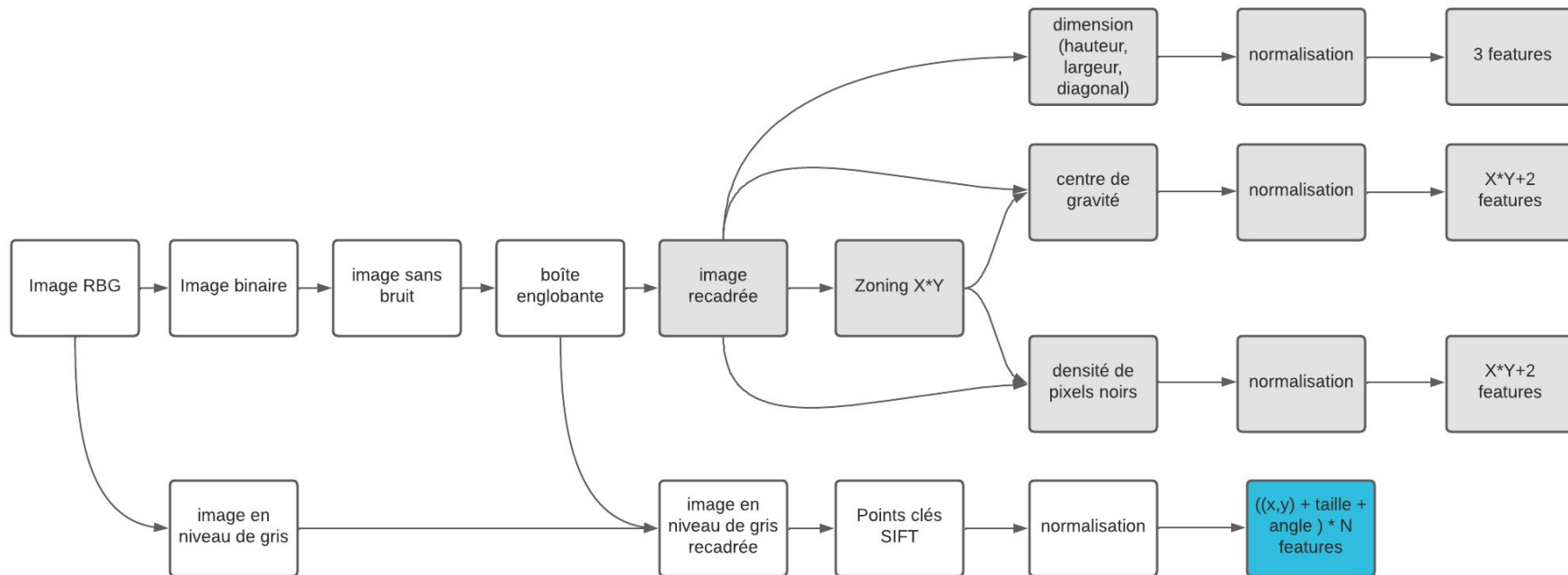
# Meilleur classifieur

Algorithme	MLP
Précisions	94.2857 %
Rappel	94.3456 %
F-Mesure	94.0151 %

=== Confusion Matrix ===

```
a b c d e f g h i j k l m n <-- classified as
25 0 0 0 0 0 0 0 0 0 0 0 0 0 | a = accident
0 21 0 0 1 3 0 0 0 0 0 0 0 0 | b = bomb
0 0 25 0 0 0 0 0 0 0 0 0 0 0 | c = car
0 0 0 23 0 0 0 0 0 0 0 2 0 0 | d = casualty
0 0 0 0 25 0 0 0 0 0 0 0 0 0 | e = electricity
0 1 0 0 0 24 0 0 0 0 0 0 0 0 | f = fire
0 0 0 0 0 0 25 0 0 0 0 0 0 0 | g = firebrigade
0 0 0 0 0 0 0 24 0 0 0 0 1 0 | h = flood
0 0 0 1 1 0 2 2 15 0 3 1 0 0 | i = gas
0 0 0 0 0 0 0 0 0 25 0 0 0 0 | j = injury
0 0 0 0 0 0 0 0 0 0 25 0 0 0 | k = paramedics
0 0 0 0 0 0 0 0 0 0 0 25 0 0 | l = person
0 0 0 0 0 0 0 0 0 0 0 0 25 0 | m = police
0 0 0 0 0 0 0 1 0 0 1 0 0 23 | n = roadblock
```

# Nouvelles features explorée : SIFT



# Extraction des features - Données non retenues

Points clés SIFT



Cas pathologiques



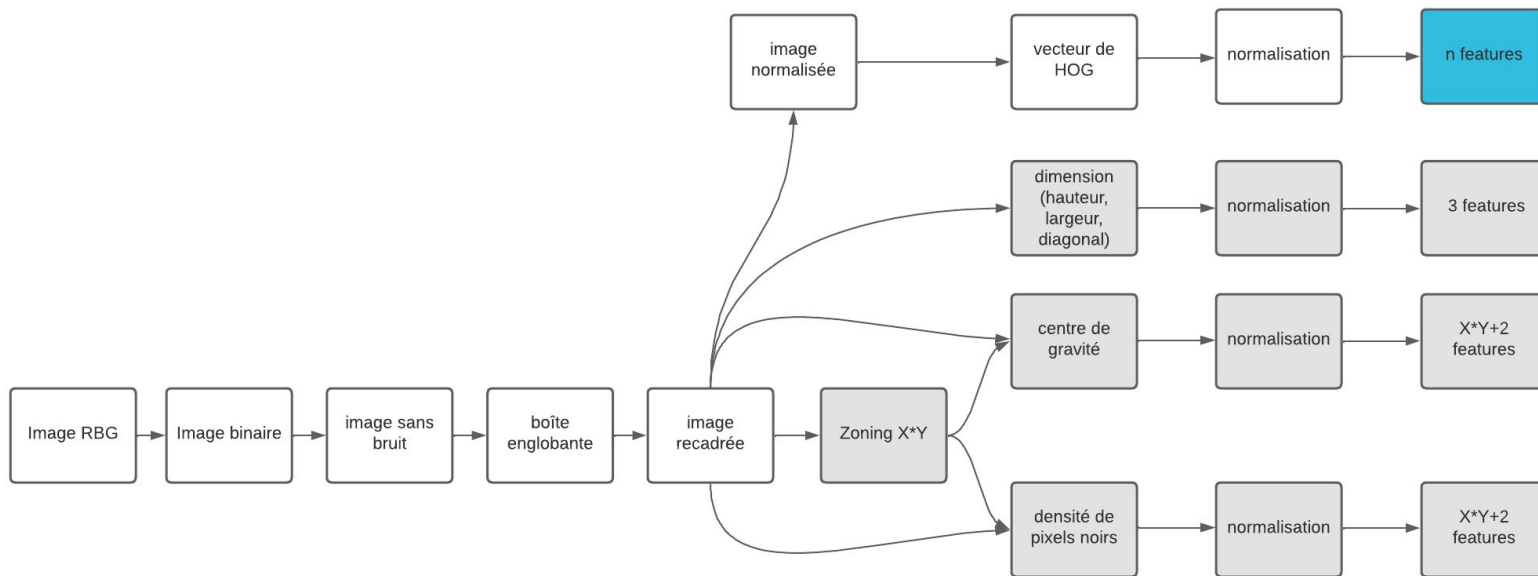
3 points clés



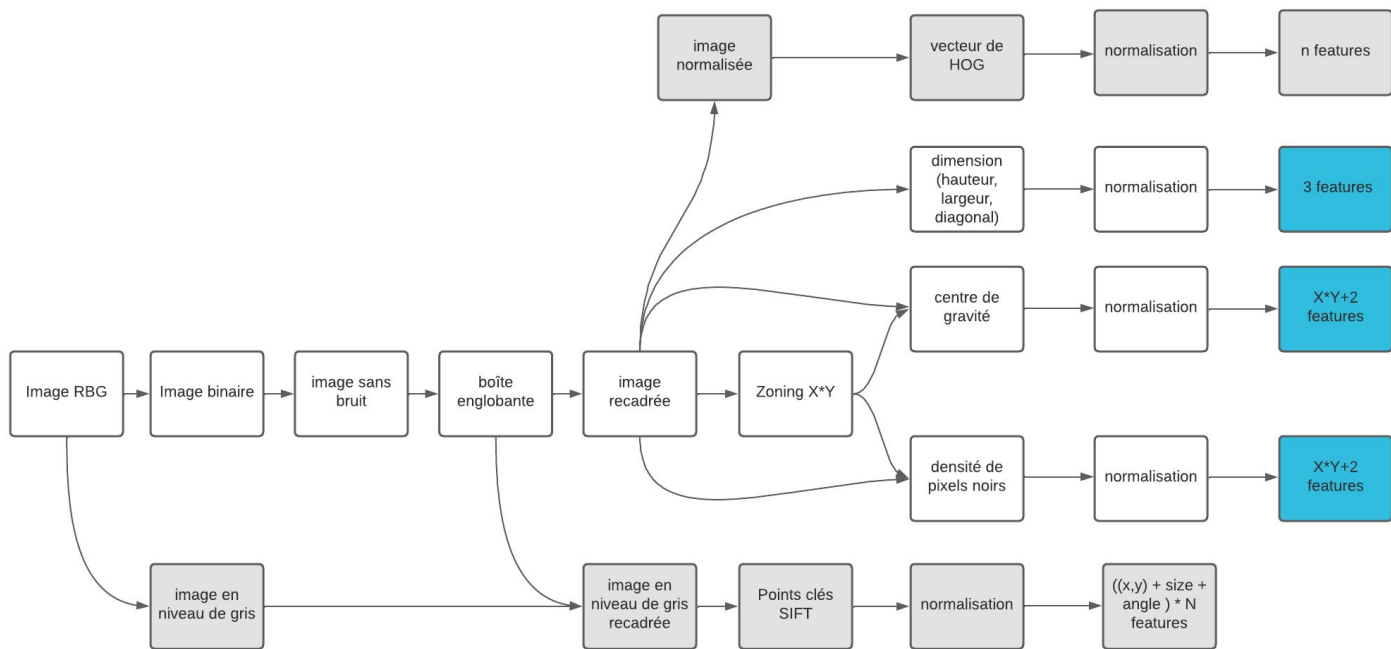
0 point clé

=> irrégularité de la feature

# Nouvelles features explorée : vecteurs de HOG



# Les résultats de notre chaîne





# Conclusion

- De nombreux features;
- Résultats avec kNN prometteur sur la base d'entraînement;
- Résultats plus faible sur les autres algorithmes => problème avec les features;
- Recherche de nouvelles features non concluante;
- Bons résultats avec le MLP (>kNN)
- Des solutions si on avait eu plus de temps:
  - Adapter la chaîne aux vecteurs de HOG
  - Utiliser AdaBoost ou un classifieur entre les meilleurs classifieurs

# Questions

