Soutenance Traitement Image & Vidéos

Application de reconnaissance automatique d’icônes

Avellaneda Tom, Cadot Firmin, Le Bizec Léandre et Moureaux Nathan

03/01/2023

Objectifs

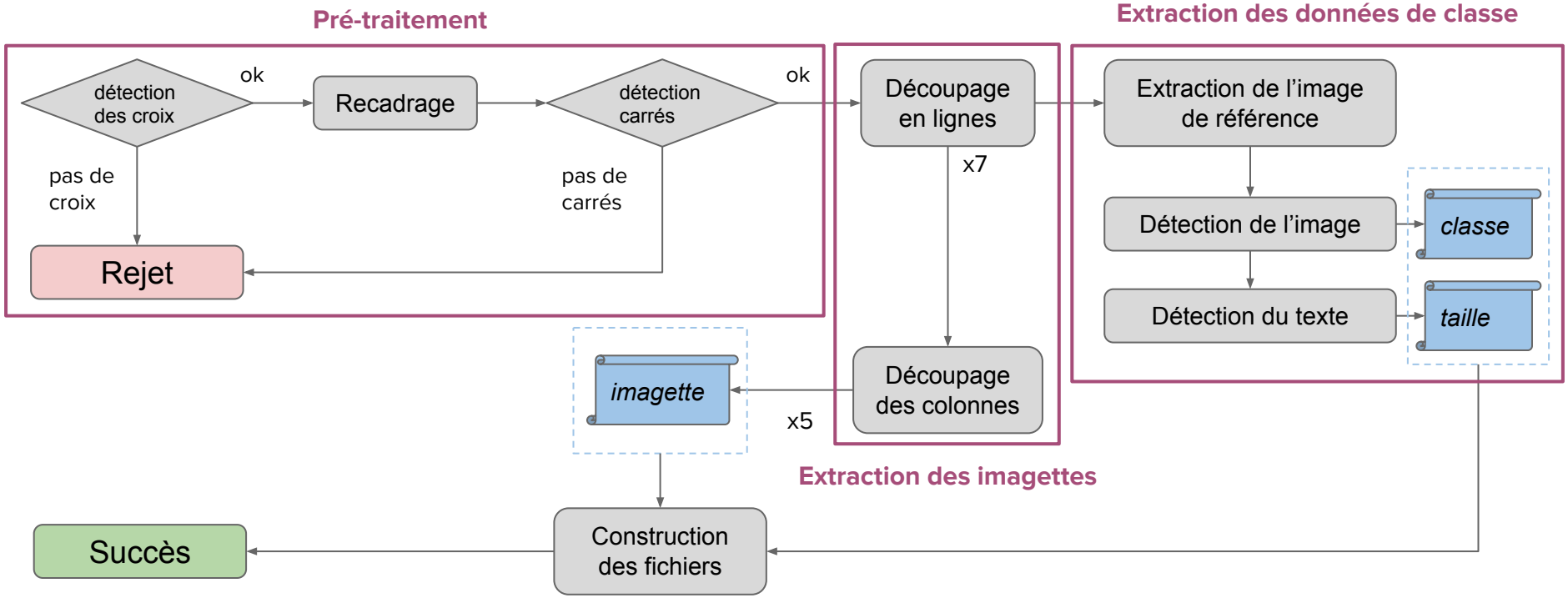
01205.png

bomb_012_05_3_2.png

```
# Firmin, Léandre, Nathan, Tom, Projet Traitement d'images 2021/2022
label bomb
form 01205
scripter 012
page 05
row 3
column 1
size medium
```

bomb_012_05_3_2.txt

La chaîne de traitement



CdT - Gestion des différents cas

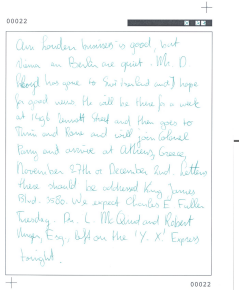
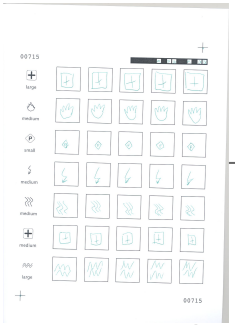


Image avec du texte

Fichier non géré
donc écarté

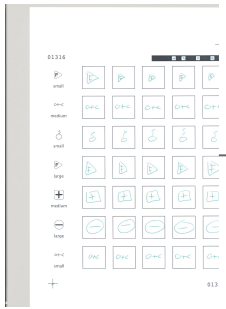


Scanne pas droit

Normalisé

Débordement

Résultat

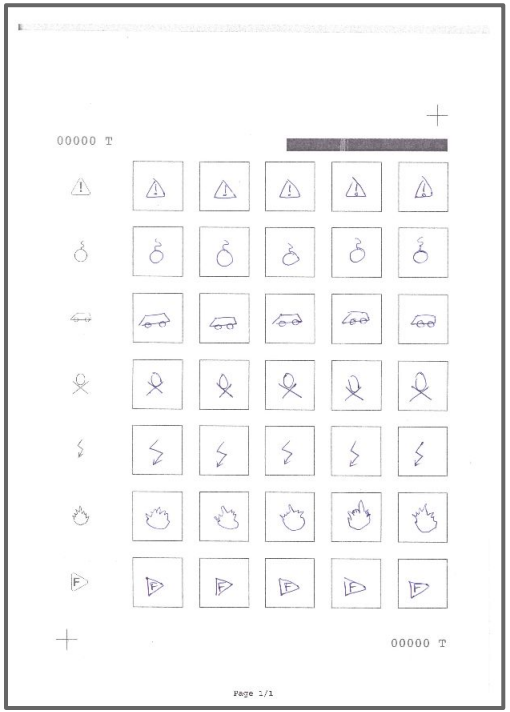


Mauvais scanne

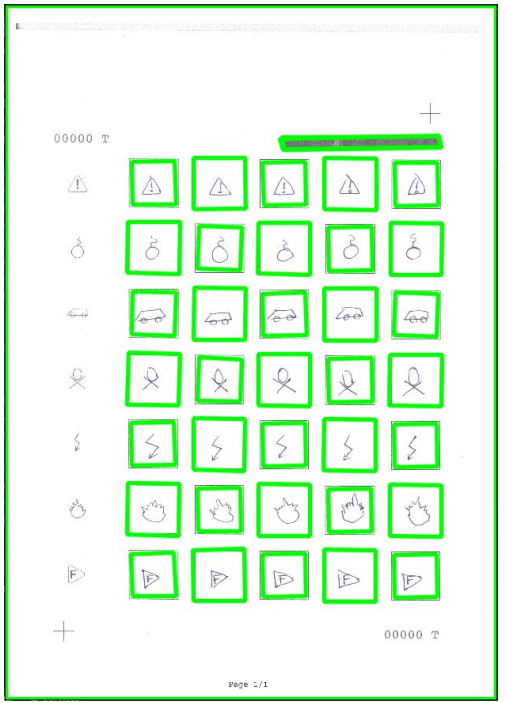
Écarté

CdT - Techniques utilisées

Détection des carrés

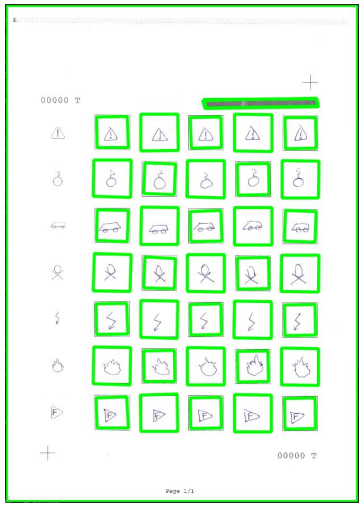


1) Détection de contours (algorithme de Canny)
-> vector<vector<Point>>
    1.1) Détection des carré -> vector<point>[4]
2) Détection de carré similaire -> comparaison
3) Traçage des carrés



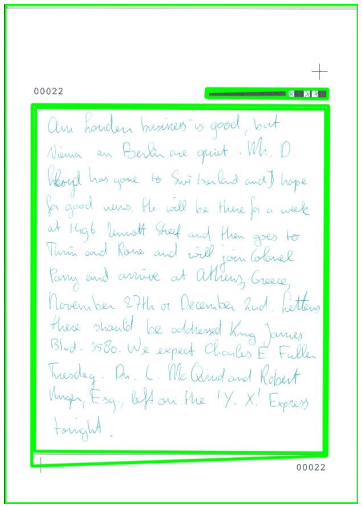
CdT - Techniques utilisées

Détection des pages de textes



Page correcte

38 carrés détectés



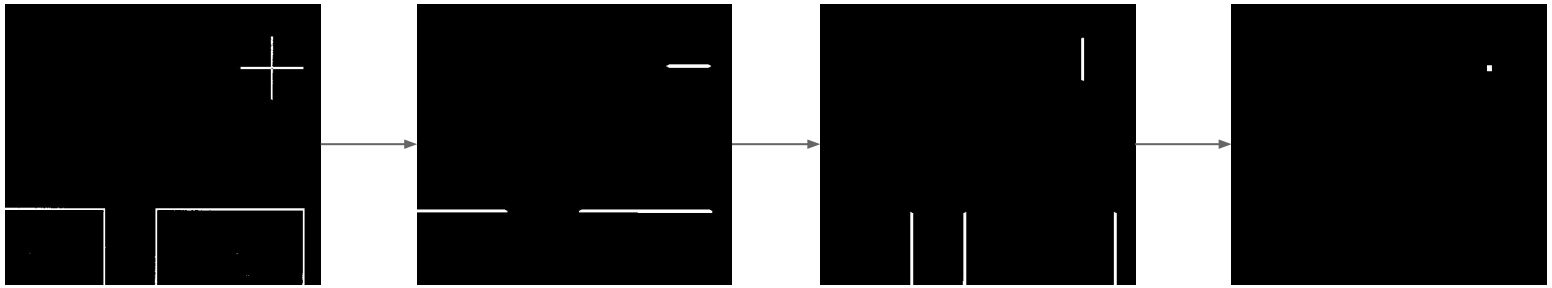
Page incorrecte

5 carrés détectés

if(nb_carrés > 10) { return true; } else { return false; }

CdT - Techniques utilisées

Détection des croix - morphologie mathématique

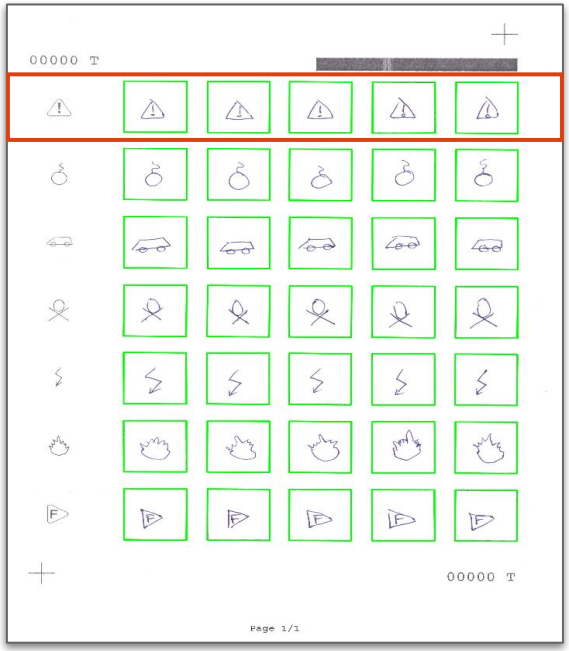


Lignes | Lignes Horizontales | Lignes Verticales | Intersections

CdT - Techniques utilisées

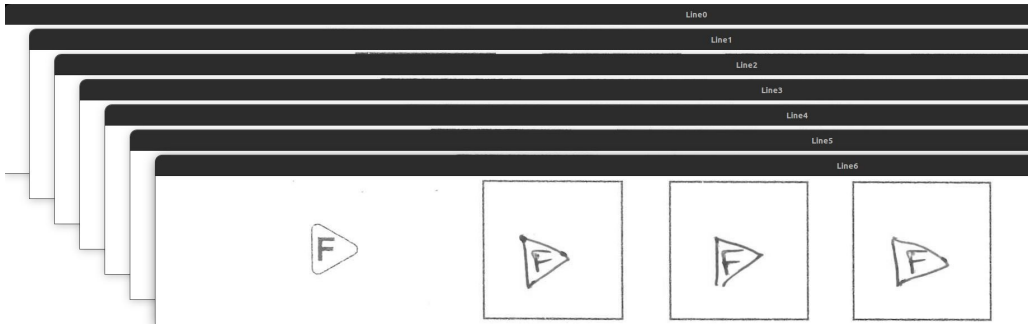
Détection des lignes d'une page

Image redressé



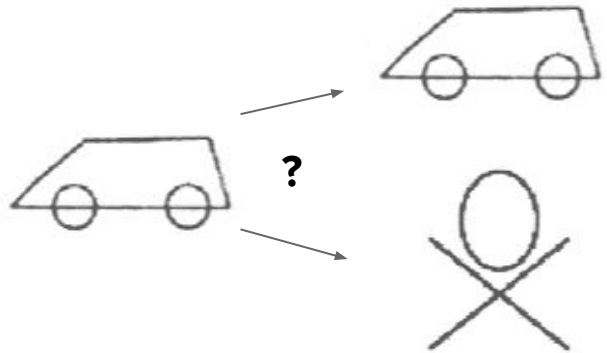
Page standardisé

- Tous les éléments sont à la même position
- On récupère les lignes par position



CdT - Techniques utilisées

Détection des images de référence - optimisation par gradient descendant



⇒ comparaison avec chaque image connue

- distance euclidienne optimale par translation

$$\min_{t \in \mathbb{R}^2} \|T(t; X) - Y\|^2$$

- algorithme de descente de gradient pour optimiser

Idem pour le texte représentant la taille

Découpage - ROI pré-déterminées

Résultats - Base d'entrainement

- Précision : 100% (échantillon 30 pages)
- Objectifs atteints pour la base d'entrainement

bomb_012_05_3_2.png

```
# Firmin, Léandre, Nathan, Tom, Projet Traitement d'images
2021/2022
label bomb
form 01205
scripter 012
page 05
row 3
column 1
size medium
```

Résultat - Base de test finale

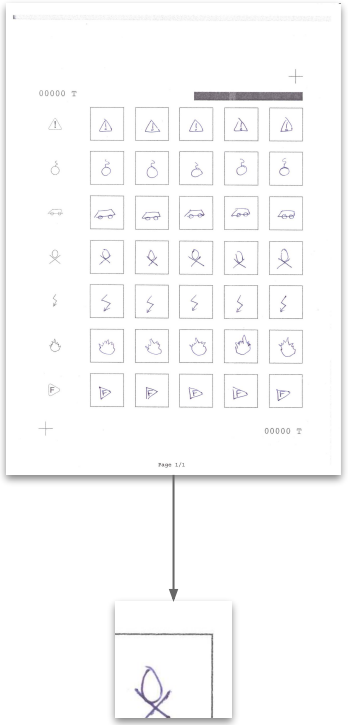
Taille de l’image différente de la base d’entraînement

↓

Problème lors du découpage des imagettes

↓

Non fonctionnel sur ce type d’image



- 0 % de précision

Format du nom des fichiers différents de la base de test

↓

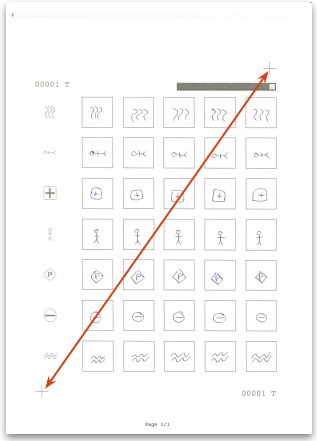
Obligation de renommer les fichiers

Conclusion

- Base d’entrainement ✅
- Base de tests ❌

⇒ Ajustement du recadrage de l’image

⇒ Amélioration de la détection des noms



Annexe

```
class SquareDetection{
public:
    SquareDetection();
    ~SquareDetection();
     bool isPointCloseEnough(Point p1, Point p2, double threshold);
     bool isSquareCloseEnough(const vector<Point>& c1, const vector<Point>& c2);
     bool isSquarePresent(const vector<Point>& c, const vector<vector<Point>>& squares);
     void findSquares( const Mat& image, vector<vector<Point> >& squares );
     void drawSquares(Mat& _image, const vector<vector<Point> >& squares);
     void printSquare(Mat image);
private:
    int thresh, N;
    const char* wndname;
};
static double angle(Point pt1, Point pt2, Point pt0);
```