

Projet Transverse

Présenté par Boris COLIN, Idriss OUGUERGOUZ, Sathya PIRABAHARAN, Dorian TURGOT, Michel WU



Sommaire

I) Présentation du projet

II) Objectifs

III) Mise en place d'un algorithme de reconnaissance

IV) Analyse des données

V) Conclusion



Vidéo présentation

Lien de la vidéo : https://www.youtube.com/watch?v=k_2Cvqb4upg



I. Présentation

- Dispositif d'analyse d'affluence
- Permet de mesurer et comprendre l'affluence à un endroit spécifique
- Adaptable et capable de fournir des analyses détaillées des données obtenues.
- Fonctionne en extrayant des données d'affluence à partir de fichiers vidéo



II. Objectifs

- Concevoir un site web sur lequel on peut déposer des vidéos à distance.
- Réaliser un algorithme de reconnaissance de personne.
- Recevoir le rapport contenant les graphes et les analyses.

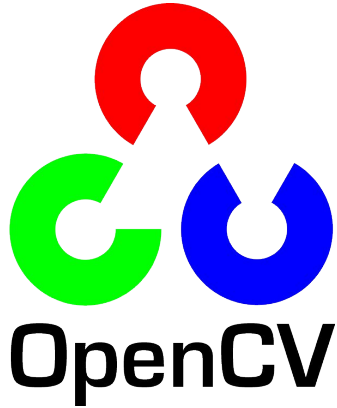


III. Mise en place d'un algorithme de reconnaissance

- Dispositif d'analyse d'affluence
- Permet de mesurer et comprendre l'affluence à un endroit spécifique
- Adaptable et capable de fournir des analyses détaillées des données obtenues.
- Fonctionne en extrayant des données d'affluence à partir de fichiers vidéo



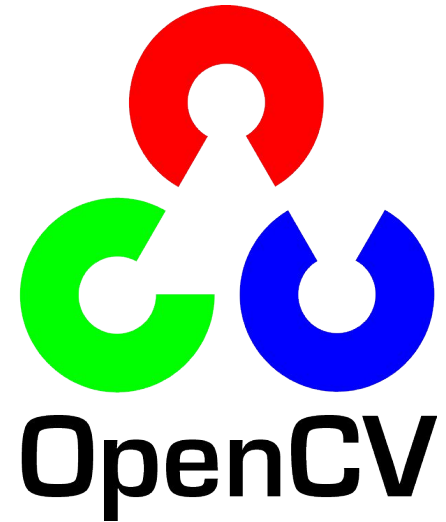
Algorithme de reconnaissance de personne





OpenCV

- **Librairie** remplie de fonctions pour faire de la **vision par ordinateur**
- Développé par **Intel** et **open-source**
- Permet de faire le **traitement d'images** afin d'exploiter les vidéos





TensorFlow

- Ensemble d'outils (**framework**) de **machine learning**
- Développé par **Google** et **open-source**
- Permet de faire la reconnaissance des personnes à partir d'une image
- Nécessite un **réseau neuronal** suffisamment entraîné



TensorFlow



Étapes de construction de l'algorithme

1. Comprendre le fonctionnement des librairies
2. Reconnaître des personnes sur une image
3. Reconnaître des personnes sur une vidéo
4. Modifier des paramètres pour fluidifier et accélérer le processus
5. Enlever le comptage des mêmes personnes
6. Rendre les données exploitables

Reconnaissance des personnes





Enlever le comptage des mêmes personnes








```
if capture:
    oldbox = []
    while framerate < frame_count:
        ret, frame = capture.read()
        if ret:
            box, label, count = cv.detect_common_objects(frame)
            totalNewPersonOnFrame = 0
            for i in range(len(box)):
                if i >= len(oldbox) or (math.sqrt((oldbox[i][0] - box[i][0]) ** 2)) > 350:
                    if label[i] == "person":
                        totalNewPersonOnFrame += round(count[i], 3)
```

Conception du site avec Flask

- Ensemble d'outils (**Framework**) web Python
- Adapter pour concevoir des sites
- Modulaire et très légère
- Simple à utiliser



Comment a-t-on utilisé Flask ?

RecoSite				
Précédent/Suivant		Présentation	Grouper	Action
		Partager	Modifier les tags	Rechercher
Nom	Date de modification	Taille	Type	
>  __pycache__	aujourd'hui à 20:50	--	Dossier	
 analyse.py	aujourd'hui à 19:40	1 ko	Python source	
 app.py	5 décembre 2022 à 15:25	265 octets	Python source	
 main.py	aujourd'hui à 20:49	1 ko	Python source	
>  static	aujourd'hui à 20:50	--	Dossier	
>  templates	aujourd'hui à 20:50	--	Dossier	
 video_traitement.py	aujourd'hui à 19:07	1 ko	Python source	

Comment a-t-on utilisé Flask ?

app.py

```
from flask import Flask

UPLOAD_FOLDER = 'static/uploads/'
EXTENSIONS = {'mp4'}

app = Flask(__name__)
app.secret_key = "secret key"
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

# 16 * 1000 * 1000 = 16 Mo
app.config['MAX_CONTENT_LENGTH'] = 16 * 1024 * 1024 * 1024
```

Comment a-t-on utilisé Flask ? (suite)

main.py

```
from flask import Flask, flash, request, redirect, url_for, render_template
from werkzeug.utils import secure_filename
from app import app

from video_traitement import *
from analyse import *

@app.route('/')
def menu():
    return render_template("index.html")

@app.route('/', methods=['POST'])
def upload_file():
    if request.method == 'POST':
        pass
        f = request.files['file']
        if f.filename == '':
            flash('Pas de vidéo sélectionnée')
            return redirect(request.url)
        f.save(secure_filename("video.mp4"))
        return redirect(url_for("traitement"))

@app.route('/index.html')
def menu2():
    return render_template("index.html")

@app.route('/stats.html')
def traitement():
    people = countPeople("video.mp4")

    nb_perso = people[0]
    nb_perso_debit = nb_perso[1:]
    framerate = people[3]

    affichage(nb_perso_debit, framerate, 0)

    result = open("static/analyse.png", "r")

    return render_template("stats.html", result=result)

if __name__ == '__main__':
    DEBUG = True
    HOST = '0.0.0.0'
    app.run(debug=DEBUG, host=HOST, port=80)
```


Adaptation de Flask en HTML

index.html

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="shortcut icon" href="../static/logo.png">
    <link href="{{ url_for('static', filename='style.css') }}" rel="stylesheet" type="text/css">
    <title> Recon0ne Accueil </title>
  </head>
  <body>
    <header>
      
    </header>
    <main>
      <div id="rcorners1">
        <h1> Quelle vidéo voulez-vous traiter ?</h1>
        <p>
          {% with messages = get_flashed_messages() %}
          {% if messages %}
            <ul>
              {% for message in messages %}
                <li>{{ message }}</li>
              {% endfor %}
            </ul>
          {% endif %}
          {% endwith %}
        </p>
        <form method="post" action="/" enctype="multipart/form-data">
          <dl>
            <input type="file" name="file" autocomplete="off" accept="video/mp4" required>
          </dl>
          <input type="submit" class="button-3" onclick="loading()" value="Traiter">
        </form>
        
      </div>
    </main>
    <footer>
      <div class="navbar">
        <ul>
          <li> <a href="#">Vidéo du projet</a></li>
          <li> <a href="#">Rapport du projet</a> </li>
          <li> <a href="#">PPT du projet</a></li>
        </ul>
      </div>
      <script type="text/javascript" src="../static/script.js"></script>
    </footer>
  </body>
</html>
```

Adaptation de Flask en HTML (suite)

stats.html

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="shortcut icon" href="logo.png">
    <link href="{{ url_for('static', filename='../static/style.css') }}" rel="stylesheet" type="text/css">
    <title> ReconOne Statistiques </title>
  </head>
  <body>
    <header>
      
    </header>

    <main>
      <div id="rcorners1">
        <h1>Statistiques de la vidéo</h1>
        {% if result %}
          
        {% endif %}
      </div>
    </main>

    <footer>
      <div class="navbar">
        <ul>
          <li> <a id="revenir" href="index.html"> Revenir à l'accueil </a> </li>
        </ul>
      </div>
    </footer>
  </body>
</html>
```



IV. Analyse des données

1) Type de données :

Liste du nombre de nouvelles personnes recensées à chaque intervalle pondéré par la fiabilité de la reconnaissance.

Début de la vidéo en heure, et durée entre chaque intervalle de travail.

2) Calcul du débit :

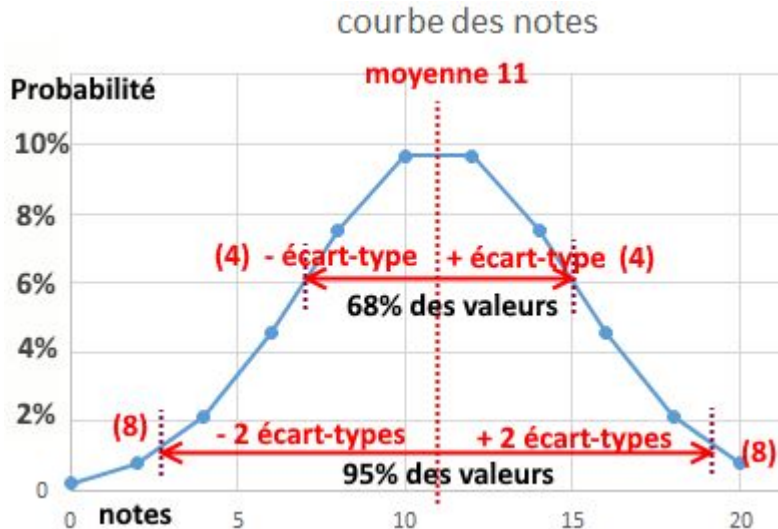
Le débit est l'écoulement continu des entités par unité de temps. Récupérer un graphique détaillant l'évolution du débit peut avoir de nombreuses applications.

On peut évaluer l'efficacité de l'évacuation des gens, ou savoir s'il y a un pic d'affluence quand le débit augmente fortement puis diminue.



3) Traitement en tant que série statistique :

Ici, on va considérer que chaque nouvel élément de la liste des nouvelles personnes appariées est une variable aléatoire. Cela fait qu'on peut analyser la liste comme une série statistique, ce qui inclut des calculs d'espérance et d'écart-type.

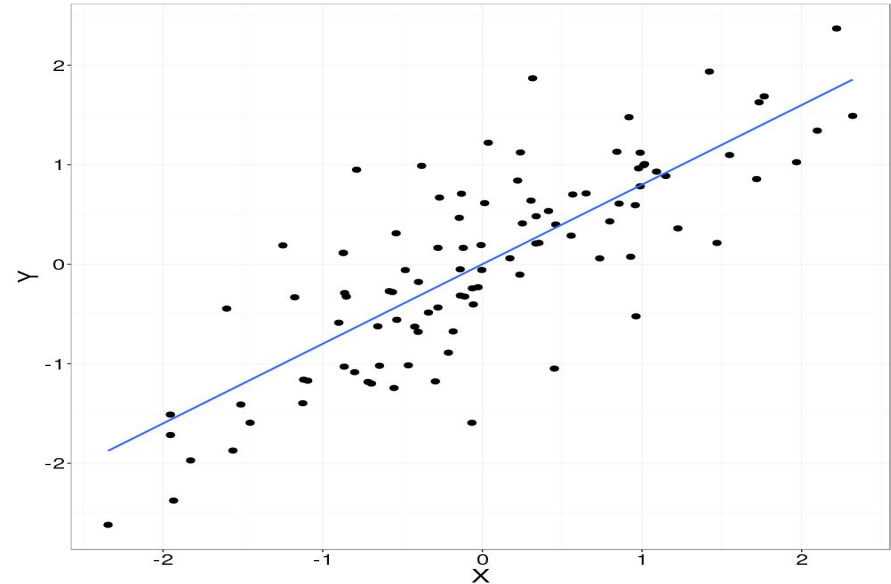


valeur moyenne

$$\text{écart-type} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \overline{x})^2}$$

4) Analyses plus poussées:

Du fait que la liste ne peut être traitée comme rien de plus qu'une série statistique, il est difficile de faire des analyses plus approfondies des données. Il devient impossible d'effectuer des régressions lors de l'écart à des lois de probabilités connues.



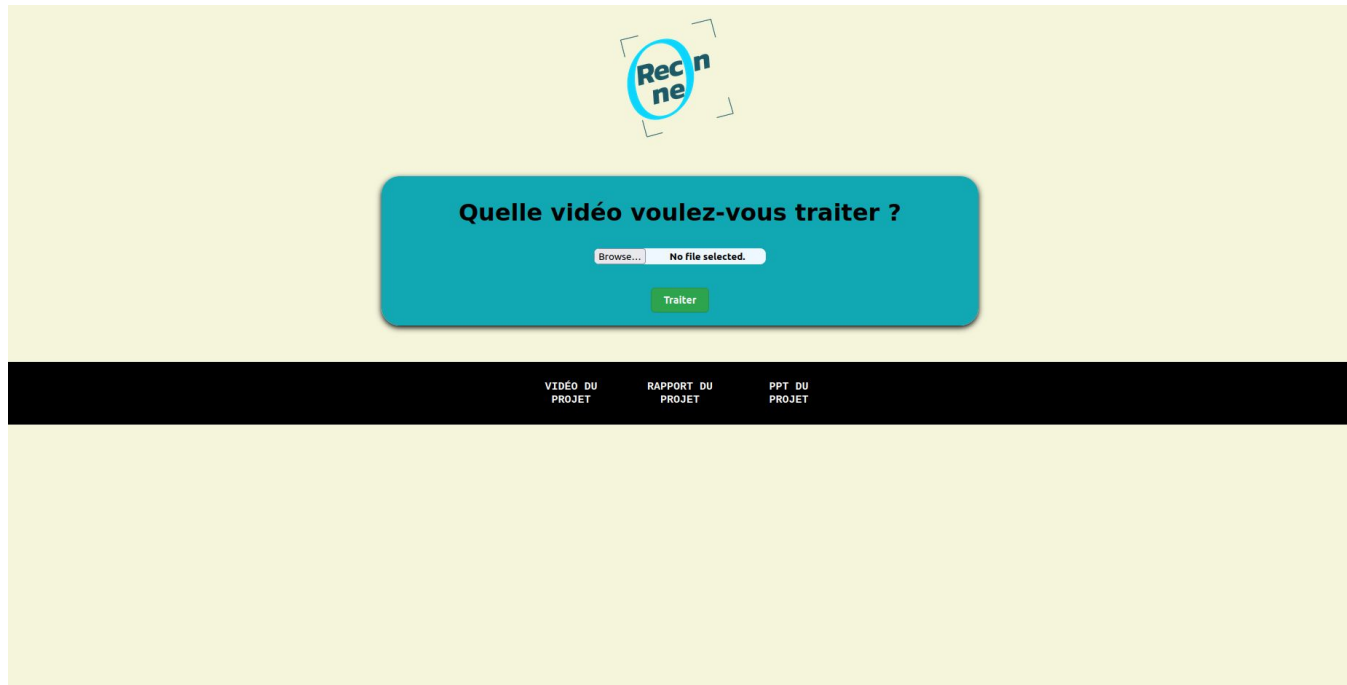


Conclusion

- Objectifs du cdc : OK
- Améliorations personnelles : ↗Python, ↗Travail d'équipe
- Perspective d'avenir
- Résultats (voir diapo suivantes)



Résultat du site



Résultat du site (suite)

