

Kevin LE GRAND  
Harold PETIARD  
Antoine ANCELIN

# DETECTION D'INCENDIE



SIMPLON.CO

**ISEN**

ALL IS DIGITAL!  
**QUEST**



yncréa

# CONTEXTE :

Les services de secours souhaitent améliorer leur capacité de réponse en cas d'incendie, et pour cela ils ont exprimé le besoin de développer une application IA de détection d'incendie. En tant que développeur IA, vous réaliserez les tâches suivantes pour répondre à leur besoin :

1. **La préparation et labélisation des données** : labéliser la DataSet fournie. (possibilité de rajouter certaines images, de faire de l'augmentation de données)
2. **Transfer Learning** : réaliser un Transfer Learning sur l'architecture de YOLOv5.
3. **Développement de l'application** : développer une application Streamlit qui sera capable de :
  - a. Charger et exécuter la détection à partir d'une image, vidéo ou d'une Webcam.
  - b. Permettre de stocker les détections dans une bdd

# **SOMMAIRE :**

**01**

**Préparation de  
données**

**02**

**Transfert  
Learning**

**03**

**Application  
streamlit**

**04**

**Démonstration**

# INTRODUCTION

- Labelisation des données.
- Entraînement d'un modèle à reconnaître un incendie (flammes et fumées) via YoloV5.
- Intégration du modèle dans une application Streamlit permettant la détection d'un incendie depuis une image, une vidéo ou bien une webcam.
- Enregistrement en base de données et consultation depuis l'application.



# 01

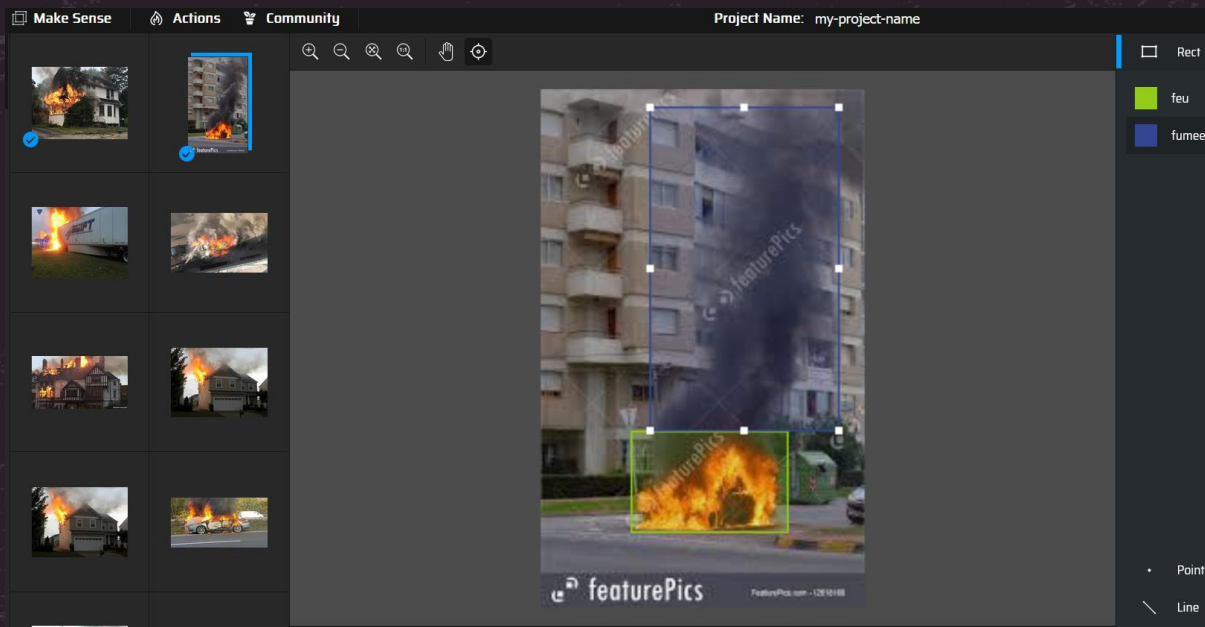
## La Labélisation :



# CONCEPTS

Labélisation : sélectionner manuellement, sur des images, les zones où apparaissent des flammes et/ou de la fumée.

Outils : makesense.ai





**02**

**Le Transfert Learning:**



# CONCEPTS

Transfert Learning : utiliser un modèle déjà entrainer, dans notre cas, YoloV5, afin de continuer son entrainement en lui fournissant les images précédement labélisées.

Utilisation des poids de YOLOv5 pour accélérer l'entrainement avec les nouvelles images labélisées.

Possibilité de réutilisation des poids les plus performant lors de l'entrainement pour réaliser nos détections.





# Déroulement de l'entraînement

- Uploader YoloV5 sur son ordinateur depuis le dépôt github
- Charger nos image dans un dossier datasets
- Créer un fichier .yaml pour définir le chemin de nos images labélisées et les labels
- Utiliser GoogleColab pour accélérer l'entraînement
- Deux solutions possible pour paramétrer l'entraînement :
  - Modifier directement le train.py
  - Ajouter les paramètres dans la ligne de commande
- Faire un test sur une ou plusieurs images

```
!python yolov5/train.py --data yolov5/data/feu.yaml --epochs 15 --batch-size 16
```

```
Validating yolov5/runs/train/exp3/weights/best.pt...
```

```
Fusing layers...
```

```
Model summary: 157 layers, 7015519 parameters, 0 gradients
```

Class	Images	Instances	P	R	mAP50
all	90	178	0.664	0.635	0.648
feu	90	123	0.61	0.805	0.716
fumee	90	55	0.719	0.465	0.581

```
!python yolov5/detect.py --weights yolov5/runs/train/exp3/weights/best.pt --source Donnees/test
```



# Amélioration du modèle :



Amélioration du modèle possible :

- Ajouts de plus d'image avec de la fumée
- Utiliser un autre modèle pré-entraîné de YOLOv5 (ici on a utilisé yolov5s.pt)

En utilisant le modèle pré-entraîné supérieur de yolov5 (yolov5m.pt) les performances sont effectivement améliorées:



```
!python yolov5/train.py --data yolov5/data/feu.yaml --weights yolov5m.pt --epochs 20 --batch-size 16
```

## yolov5s

Validating yolov5/runs/train/exp3/weights/best.pt...

Fusing layers...

Model summary: 157 layers, 7015519 parameters, 0 gradients

Class	Images	Instances	P	R	mAP50
all	90	178	0.664	0.635	0.648
feu	90	123	0.61	0.805	0.716
fumee	90	55	0.719	0.465	0.581

## yolov5m

Validating yolov5/runs/train/exp5/weights/best.pt...

Fusing layers...

Model summary: 212 layers, 20856975 parameters, 0 gradients, 47.9 GFLOPs

Class	Images	Instances	P	R	mAP50
all	90	178	0.784	0.707	0.73
feu	90	123	0.776	0.862	0.843
fumee	90	55	0.792	0.553	0.617

**03**

**L'application streamlit**

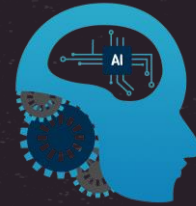


# STRUCTURE DE L'APPLICATION :

1. Insertion d'une source dans l'application.
2. Réalisation de la détection par l'IA.
3. Affichage de la détection dans l'application et stockage en base de données.
4. Création d'un onglet archive permettant la visualisation de toutes les détections effectuées dans l'application via l'archivage en base de données.



MOBILITY



**04**

**Démonstration**





# Remerciements – Liens utiles

Merci à Charley et David pour la labélisation des images

Merci à Sylvestre Apetcho pour son tuto sur yoloV5:  
<https://youtu.be/loFtqLbtk6A>

Site pour la labélisation : <https://www.makesense.ai>

