

08.05.2023

# Plates

## Traitement d'images II

Owen Gombas, David Darmanger

# Sommaire

- Problématique
- Méthodologie
  - Pipeline complet
- Résultats
- Discussion des résultats
- Démonstration
- Limitations et perspectives
- Conclusion

# Problématique

- Détecter les numéros d'identification de plaques d'immatriculation de véhicules
- Photos par une caméra de surveillance
  - Même angle
  - Même caméra → plus d'algorithme de traitement d'image (moins de Deep Learning)
- Plaques vietnamiennes → traitement adapté aux spécificités de celles-ci
- Plusieurs algorithmes de traitement d'images sans Machine Learning
- Utilisation OpenCV et modèle de Transformer (TrOCR)

# Méthodologie

- Cropping sur les bords  
Paramètre: `plates.CROP = (80, 80, 80, 80)`



Cropping  
(80, 80, 80, 80)

# Méthodologie

- Transformation niveau de gris
- Application d'un filtre Gaussien pour réduire le bruit présent dans l'image.



Transformation  
en niveau de Gris



Filtre  
Gaussien

# Méthodologie

- Algorithme de Canny pour mettre en évidence les contours
- Détection des composantes connexes  
Paramètre: `plates.MIN_SIZE_CONNECTED_COMPONENT = 120`



Canny



Composants  
connectés



# Méthodologie

- Analyse et sélection des composantes connexes  
Paramètre: `plates.ASPECT_RATIO_MIN_MAX = (0.3, 1.8)`  
Paramètre: `plates.HEIGHT_MIN_MAX = (40, 140)`  
Paramètre: `plates.WIDTH_MIN_MAX = (40, 140)`
- Mise en évidence des plaques potentielles
- Recadrage sur les plaques potentielles



Région



Plaques  
recadrées



# Méthodologie

- Binarisation de l'image pour faciliter la reconnaissance des caractères
- Division des plaques en 2 parties (haut/bas) en utilisant une signature
  - Modèle fonctionne que sur 1 ligne de texte





# Méthodologie

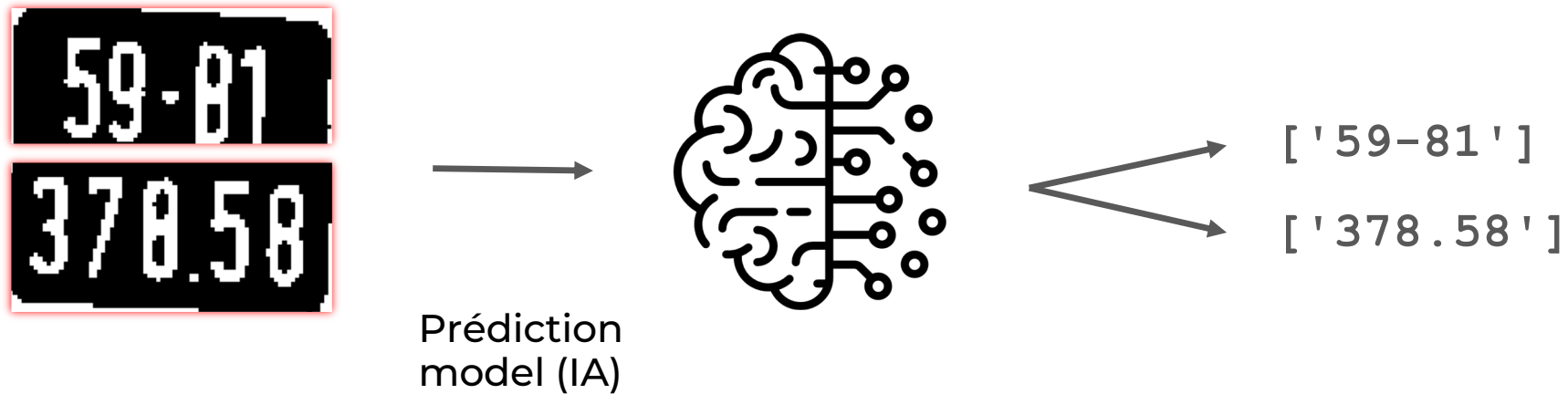
- Division des plaques en 2 parties (haut/bas) en utilisant une signature
  - Séparation = ligne avec le plus de pixels noirs
  - Marge de bordure

Paramètre: `plates.SPLIT_CROP=(15)`

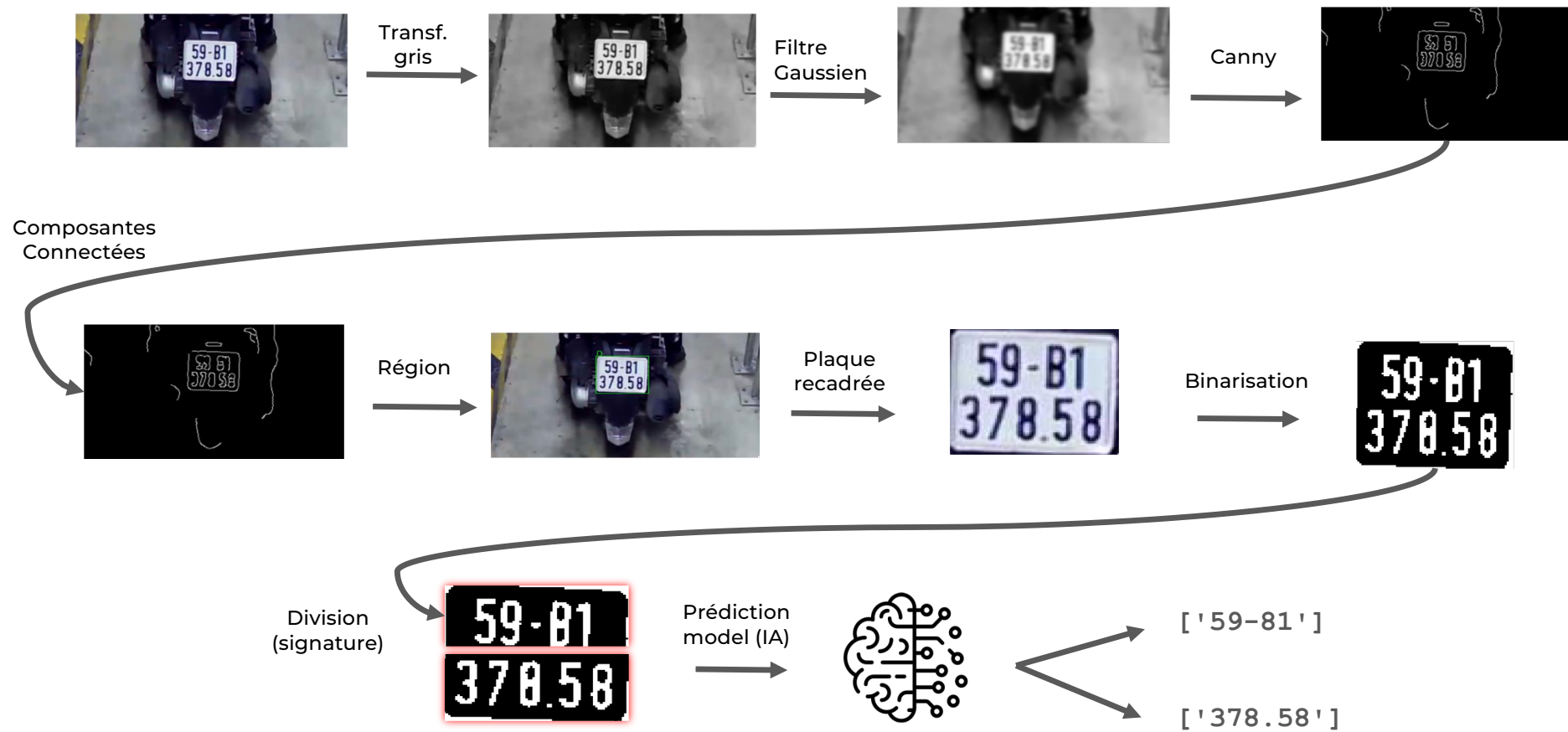


# Méthodologie

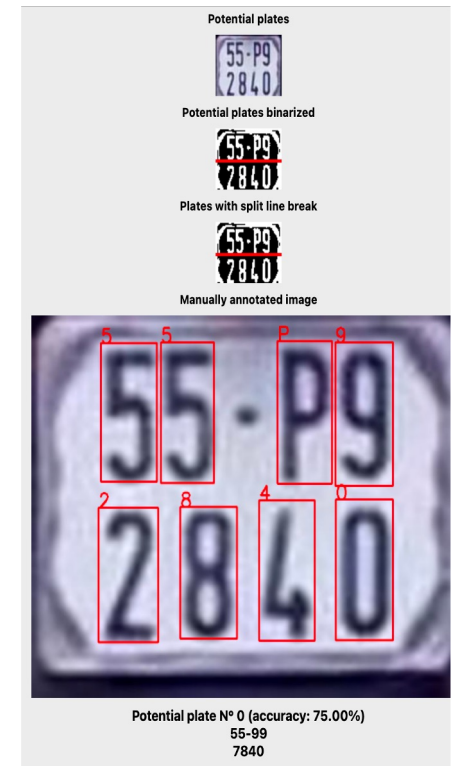
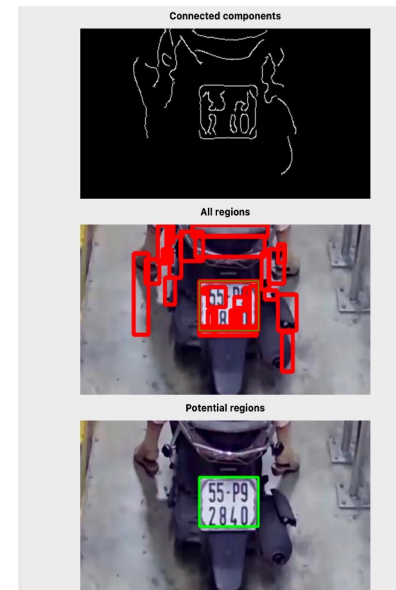
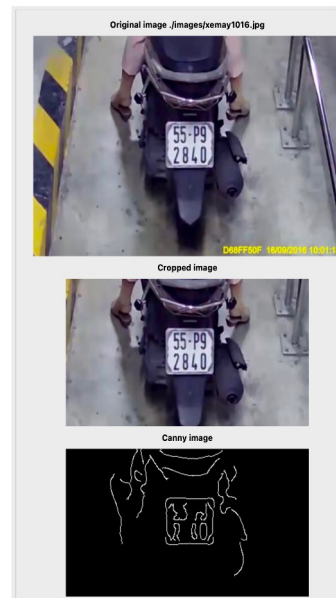
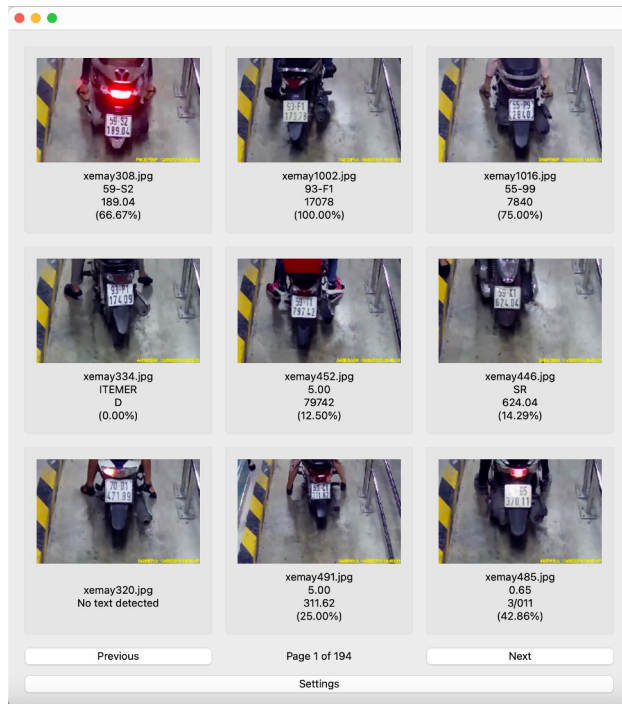
- Détection (IA) des caractères du haut et du bas de la plaque séparément



# Pipeline complet depuis le rognage initial



# Résultats



# Discussion des résultats

- Réalisation des tests sur un ensemble des photos en utilisant notre pipeline de traitement d'images
- Résultats satisfaisants
  - détection des plaques d'immatriculation
  - reconnaissance correcte des caractères des plaques détectées
- Quelques erreurs
  - pipeline n'a pas détecté des plaques
  - les paramètres ne sont pas adaptés à un cas spécifique
  - le modèle de reconnaissance de caractères a commis des erreurs
    - conditions d'éclairage difficiles, caractères mal alignés sur la plaque, ...

# Démonstration

# Limitations et perspectives

## Limitations

- Modèle peut détecter que les textes sur une seule ligne
- Méthode de détection des plaques adaptée aux plaques thaïlandaises capturées sous un même angle et par la même caméra

## Perspectives

- Algorithmes de détection de plaques plus robustes
- Étendre la méthode de détection des plaques
  - différents types de plaques d'immatriculation
  - différents angles de prise de vue
- Réduction d'effet « d'italique » sur les plaques



# Conclusion

- Solution efficace pour la détection des plaques d'immatriculation à partir d'images prises par une caméra de surveillance
  - utilisations d'algorithmes et méthodes de traitement d'images
- Extraction des caractères d'identification des plaques avec précision
- Quelques limitations concernant la détection et le modèle
  - approche peut être améliorée et généralisée à d'autres situations et problèmes similaires



Merci de votre attention !

Questions ?

# Références

- *NNDAM, Vietnamese-License-Plate-Generator*  
<https://github.com/NNDam/Vietnamese-License-Plate-Generator>
- *Winter2897, Real-time-Auto-License-Plate-Recognition-with-Jetson-Nano*  
<https://github.com/winter2897/Real-time-Auto-License-Plate-Recognition-with-Jetson-Nano>
- *Microsoft, trocr-base-printed*  
<https://huggingface.co/microsoft/trocr-base-printed>
- *Symisc, sod*  
<https://github.com/symisc/sod>  
<https://sod.pixlab.io/articles/license-plate-detection.html>

# Références (illustrations)

Flaticon :

[https://www.flaticon.com/free-  
icon/brain\\_9240330?term=machine+learning&page=1&position=3&origin=search&  
related\\_id=9240330](https://www.flaticon.com/free-icon/brain_9240330?term=machine+learning&page=1&position=3&origin=search&related_id=9240330)