



## Stage M2

**Etat de l'art logiciel traitement vidéo**

Benjamin Serva  
Master 2 IMAGINE  
Université de Montpellier

20 janvier 2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Fonctionnalités souhaité</b>	<b>4</b>
2.1	Volet 1 . . . . .	4
2.2	Volet 2 . . . . .	4
2.2.1	Premier temps . . . . .	4
2.2.2	Second temps . . . . .	4
<b>3</b>	<b>Different Logiciel pour de la lecture/traitement de vidéo</b>	<b>5</b>
3.1	VLC . . . . .	5
3.1.1	PLateforme . . . . .	5
3.1.2	Détail . . . . .	5
3.1.3	Langage de programmation utilisé pour la conception . . . . .	5
3.2	QuickTime . . . . .	6
3.2.1	Plateforme . . . . .	6
3.2.2	Détail . . . . .	6
3.2.3	Langage de programmation utilisé pour la conception . . . . .	6
3.3	Screen . . . . .	6
3.3.1	Plateforme . . . . .	6
3.3.2	Détail . . . . .	6
3.3.3	Langage de programmation utilisé pour la conception . . . . .	7
3.4	IINA . . . . .	7
3.4.1	Plateforme . . . . .	7
3.4.2	Détail . . . . .	7
3.4.3	Langage de programmation utilisé pour la conception . . . . .	7
3.5	MPV . . . . .	8
3.5.1	Plateforme . . . . .	8
3.5.2	Détail . . . . .	8
3.5.3	Langage de programmation utilisé pour la conception . . . . .	8
3.6	DaVinci Resolve . . . . .	9
3.6.1	Plateforme . . . . .	9
3.6.2	Détail . . . . .	9
3.6.3	Langage de programmation utilisé pour la conception . . . . .	9
3.7	Comparaison . . . . .	10
<b>4</b>	<b>Article</b>	<b>10</b>
4.1	KinoAI . . . . .	10
4.1.1	Contexte . . . . .	10
4.1.2	Application . . . . .	10
4.1.3	Retour . . . . .	10
<b>5</b>	<b>Bibliothèques INA</b>	<b>11</b>
5.1	inaSpeechSegmenter . . . . .	11
5.1.1	Language . . . . .	11
5.1.2	Pré-requis . . . . .	11
5.1.3	Fonctionnalités Proposées . . . . .	11
5.2	inaFaceAnalyser . . . . .	11
5.2.1	Language . . . . .	11
5.2.2	Pré-requis . . . . .	11
5.2.3	Fonctionnalités Proposées . . . . .	11
5.3	speaking-clock-detection . . . . .	12

5.3.1	Language . . . . .	12
5.3.2	Pré-requis . . . . .	12
5.3.3	Fonctionnalités Proposées . . . . .	12
<b>6</b>	<b>Méthodes</b>	<b>12</b>
6.1	Transcription d'un audio en texte . . . . .	12
6.1.1	Alternative 1 : SpeechRecognition . . . . .	12
6.1.2	Alternative 2 : Google Cloud Speech-to-Text . . . . .	13
6.2	Gestion et extraction d'informations intégrées dans les fichiers vidéo .	13
6.3	Reconnaissance de texte dans les vidéos : Exploitation d'OCR (Re-connaissance Optique de Caractères) . . . . .	14
6.4	Segmentations de scènes . . . . .	15
6.4.1	Shot Boundary Detection (SBD) . . . . .	15
6.5	YOLO : You Only Look Once . . . . .	15
6.5.1	Détection d'objets et identification . . . . .	15
6.5.2	Poursuite de cible . . . . .	16
6.6	Détection de Visage . . . . .	16
6.7	Optimisation des Performances . . . . .	18
<b>7</b>	<b>Définition</b>	<b>18</b>
7.1	Étude iconographique . . . . .	18
7.2	Profilmique . . . . .	18
<b>8</b>	<b>Références</b>	<b>18</b>

# 1 Introduction

Cette étude de l'art a pour objectif, dans un premier temps, d'exposer les besoins liés au logiciel, puis, dans un second temps, de référencer les principaux logiciels de traitement vidéo existants ainsi que les différentes techniques essentielles au traitement des vidéos.

## 2 Fonctionnalités souhaité

### 2.1 Volet 1

- interface ergonomique et intuitive inspirée de Screen
- lecture de plusieurs vidéos en simultanément et synchroniquement
- zoom
- affichage d'image fixes, texte
- lecture de son
- système de calque/grille
- outil de segmentation
- interface d'annotation manuelle
- capture images fixe et capture sonore
- découpage et encodage de segment source vidéo en format commun
- recherche **iconographique**\* en ligne

### 2.2 Volet 2

#### 2.2.1 Premier temps

- détection intelligente des plans.
- types de transitions.
- échelles de cadrage.
- types de mouvements optiques.
- paramètres colorimétriques.

#### 2.2.2 Second temps

Reconnaissance automatisée de contenus "**profilmiques**"\* spécifiques (types de situation, comportements d'objets ou d'acteurs, d'objets sonores).

### 3 Différent Logiciel pour de la lecture/traitement de vidéo

#### 3.1 VLC

##### 3.1.1 PLateforme

Multi plateforme et disponible en 69 langues.

##### 3.1.2 Détail

VLC est un lecteur multimédia libre et open source, reconnu pour sa compatibilité étendue avec la plupart des formats audio et vidéo sans nécessiter de codecs supplémentaires. Il permet également la conversion de fichiers multimédias et l'extraction de la piste audio d'une vidéo.

Il prend en charge la gestion avancée des sous-titres, incluant le chargement automatique, la synchronisation et la personnalisation de l'affichage.

L'un des points forts de VLC est sa robustesse : il peut lire des fichiers multimédias incomplets, endommagés ou en cours de téléchargement. Il supporte également la lecture en streaming à partir de flux réseau, y compris les protocoles HTTP, RTP, RTSP et HLS.

VLC offre une large gamme de filtres et d'effets vidéo, tels que la distorsion, la rotation, l'inversion, le désentrelacement, l'ajustement des couleurs, l agrandissement et le redimensionnement. Il propose aussi des effets audio, comme l'égalisation et la spatialisation du son.

Enfin, VLC dispose d'une API qui permet de tirer parti de ses fonctionnalités, notamment pour l'intégration dans des solutions de lecture ou de traitement vidéo.

##### 3.1.3 Language de programmation utilisé pour la conception

- Language : C, C++ et objective-C.
- Interface graphique : Qt, ncurses et Cocoa (pour dévelloper sur Apple).

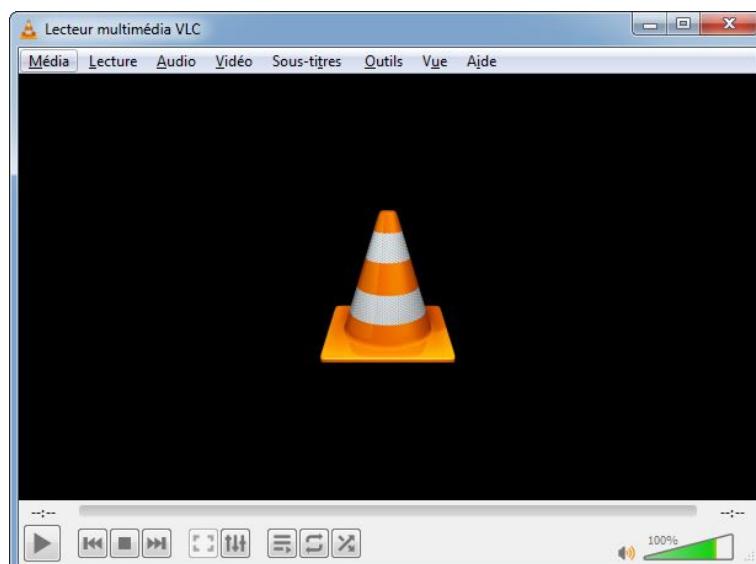


Figure 1: Interface de l'application VLC

## 3.2 QuickTime

### 3.2.1 Plateforme

Disponible uniquement sur Windows et macOS.

### 3.2.2 Détail

QuickTime est un lecteur multimédia développé par Apple, principalement utilisé pour la lecture de fichiers multimédias sur macOS et Windows. Il prend en charge une large gamme de formats audio et vidéo, notamment MOV, MP4, AAC et AIFF. Il offre des fonctionnalités de lecture avancées, notamment la prise en charge des chapitres, des sous-titres et du réglage de la vitesse de lecture.

QuickTime est également connu pour ses capacités d'édition de base, permettant de découper, fusionner et convertir des vidéos.

Il intègre des codecs propriétaires optimisés pour l'écosystème Apple et offre une excellente qualité de lecture des fichiers multimédias.

L'API QuickTime permet aux développeurs d'intégrer ses fonctionnalités dans des applications tierces.

### 3.2.3 Langage de programmation utilisé pour la conception

- Langage : C.
- Interface graphique : Cocoa (macOS), Windows API (Windows).



Figure 2: Interface de l'application QuickTime

## 3.3 Screen

Payant → 176€.

### 3.3.1 Plateforme

Disponible sur macOS.

### 3.3.2 Détail

Screen est une application de traitement et de gestion vidéo développée par VideoVillage, spécialement conçue pour les professionnels du cinéma et de l'audiovisuel. Elle permet de visualiser, analyser et annoter des vidéos avec une grande précision.

L'une de ses fonctionnalités principales est la gestion de plusieurs pistes vidéo simultanées, ce qui permet une comparaison et une analyse côté-à-côte. Screen prend également en charge le \*\*zoom temporel et spatial\*\* pour examiner des détails fins dans les vidéos.

Il permet d'afficher et d'annoter des images fixes, de rajouter des textes, des repères et d'effectuer des découpages dans les vidéos. Son interface ergonomique facilite l'ajout d'annotations manuelles, et les outils de \*\*calque et de grille\*\* permettent un alignement précis des éléments.

L'application supporte aussi le \*\*traitement audio\*\* et la gestion des sous-titres, avec des outils de synchronisation et de personnalisation des fichiers.

Screen dispose d'une \*\*API flexible\*\* permettant une intégration dans des workflows de production vidéo complexes, ainsi que des options avancées de \*\*segmentation et de codage de segments vidéo\*\* en différents formats standards.

### 3.3.3 Langage de programmation utilisé pour la conception

- Langage : C++, Python.
- Interface graphique : Qt (multi-plateforme), Cocoa (macOS).

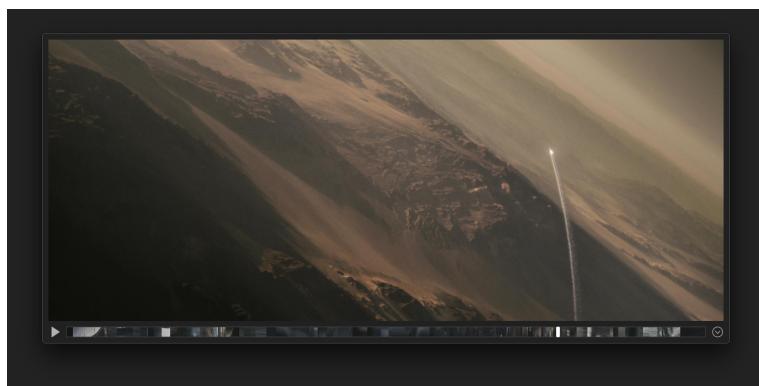


Figure 3: Interface de l'application Screen

## 3.4 IINA

### 3.4.1 Plateforme

Disponible sur macOS et basé sur le lecteur open-source mpv.

### 3.4.2 Détail

IINA est un lecteur multimédia moderne pour macOS, conçu pour offrir une expérience fluide et intuitive. Il prend en charge une large gamme de formats audio et vidéo, ainsi que le streaming. Parmi ses fonctionnalités, on trouve un mode Picture-in-Picture, l'intégration avec la Touch Bar, ainsi qu'un mode sombre natif. L'application offre également une interface utilisateur moderne et personnalisable, avec une prise en charge complète des raccourcis clavier.

### 3.4.3 Langage de programmation utilisé pour la conception

- Langage : Swift.
- Interface graphique : Cocoa (macOS).



Figure 4: Interface de l'application Iina

## 3.5 MPV

### 3.5.1 Plateforme

Disponible sur Windows, macOS et Linux.

### 3.5.2 Détail

MPV est un lecteur multimédia open-source dérivé de MPlayer et mplayer2, offrant une interface minimalist et une grande flexibilité. Il prend en charge une large variété de formats audio et vidéo, notamment MKV, MP4, AVI, AAC et FLAC. Il est particulièrement apprécié pour ses performances optimisées et son intégration avec des scripts Lua et Python, permettant une personnalisation avancée. MPV offre un support avancé des sous-titres, du rendu HDR et de l'accélération matérielle via Vulkan, OpenGL et Direct3D. Sa conception modulaire permet aux développeurs de l'intégrer facilement dans des applications tierces via son API libmpv.

### 3.5.3 Langage de programmation utilisé pour la conception

- Langage : C.
- Interface graphique : Pas d'interface native, contrôlé via CLI ou interfaces tierces.

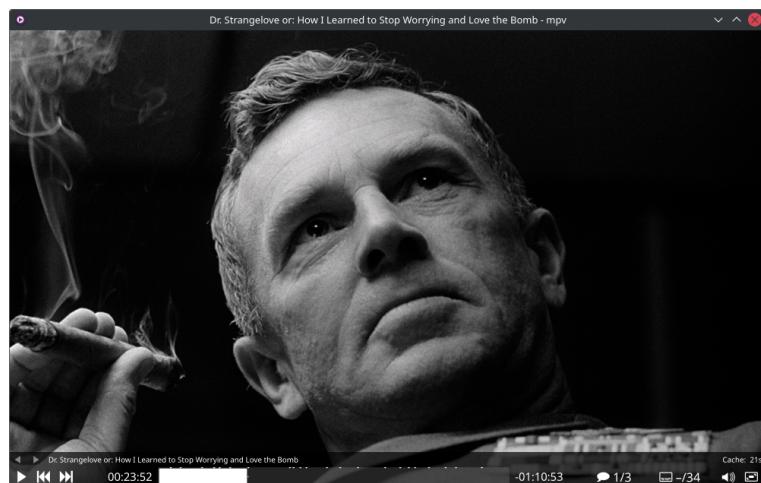


Figure 5: Interface de l'application MPV

## 3.6 DaVinci Resolve

Version Studio Payante → 354€.

### 3.6.1 Plateforme

Disponible sur Windows, macOS et Linux.

### 3.6.2 Détail

DaVinci Resolve est un logiciel professionnel de montage, d'étalonnage et de post-production vidéo développé par Blackmagic Design. Il est largement utilisé dans l'industrie cinématographique et audiovisuelle pour sa précision colorimétrique et ses outils avancés de correction des couleurs.

Il prend en charge un large éventail de formats professionnels, notamment ProRes, DNxHD, RAW et H.264/H.265.

Outre le montage non linéaire, il intègre des fonctionnalités avancées comme le suivi de mouvement, la reconnaissance faciale et des outils d'intelligence artificielle grâce à DaVinci Neural Engine.

- Reconnaissance faciale : analyse de vidéo qui sort chaque personne présente dans celle-ci avec une image. Ensuite il suffit de nommer chaque personne et on peut avoir tous les extraits avec cette personne.
- Analyse des dialogues dans une vidéo (4min pour une vidéo de 1h), affichage de tous le texte est possible d'accéder à la séquence vidéo qui contient une phrase.
- Capable de trier les sons, vidéos etc ... en catégorie.
- Détection de scène automatique

La version Studio offre des capacités supplémentaires comme l'exportation en 8K, l'optimisation du rendu GPU et la collaboration multi-utilisateurs.

### 3.6.3 Langage de programmation utilisé pour la conception

- Langages : C++, Python (pour les scripts et plugins).
- Interface graphique : Qt.



Figure 6: Interface de l'application DaVinci Resolve

### 3.7 Comparaison

Logiciel	Plateforme	Langages utilisés	API	IA	Payant
VLC	Multi-plateforme	C, C++, Objective-C	Oui	Non	Non
QuickTime	macOS, Windows	C	Oui	Non	Non
Screen	macOS	C++, Python	Oui	Non	Oui
IINA	macOS	Swift	Non	Non	Non
MPV	Multi-plateforme	C	Oui	Non	Non
DaVinci Resolve	Multi-plateforme	C++, Python	Oui	Oui	Oui/Non

Table 1: Comparaison des logiciels de traitement vidéo

## 4 Article

### 4.1 KinoAI

#### 4.1.1 Contexte

C'est un logiciel novateur qui utilise l'IA pour analyser des captations vidéo fixes de répétitions.

#### 4.1.2 Application

Détection de personnages et calcul de cadrage automatique les plus pertinent.  
Pour ce qui est de la détection de personnage la librairie OpenPose a été utilisé.  
Pour le découpage des algorithme d'optimisation temporelle ont été utilisé.

Problématique : Pour 1h de captation on obtient 10h de découpage.  
Donc il fallait limiter au maximum les prise de vue, l'application a été complètement retravaillé pour intégrer une version client-serveur avec un système de visualisation en ligne pour éviter une limite de temps sur les données fournies/traitées.

Système d'annotation avec deux modes:

- Mode Table : notation simplement importé et associé, la notation est mise en rouge quand l'action associé apparaît, notation cliquable pour visionner la scène associé.
- Mode Planche : choix des cadres en fonction des notations, séquence découpé en fonction des notations, plan large et plan choisi par l'ingénieur en fonction de la note.

#### 4.1.3 Retour

Prise en main compliquée et temps de traitement trop élevé.



Figure 7: Interface de l'application KinoAI

## 5 Bibliothèques INA

### 5.1 inaSpeechSegmenter

#### 5.1.1 Language

Elle est développée en python.

#### 5.1.2 Pré-requis

**ffmpeg** pour décoder n'importe quel type de format.

#### 5.1.3 Fonctionnalités Proposées

1. Détection de genre du locuteur : homme ou femme (modèle de classification optimisé pour la langue française car entraîné dessus).
2. Capable de dire si un humain chante ou parle. Parole sur bruit ou parole sur musique sont étiquetées comme "Parole" et la voix chantée est étiquetée comme musique.

### 5.2 inaFaceAnalyser

#### 5.2.1 Language

Elle est développée en python.

#### 5.2.2 Pré-requis

**cmake**, **FFmpeg** et **libgl1-mesa-glx**.

#### 5.2.3 Fonctionnalités Proposées

Utilise un modèle de classification basé sur l'architecture ResNet50.

1. Prédiction de l'âge et du sexe à partir des visages détectés dans des images ou des vidéos.

2. Exportation des résultats sous différents formats, notamment des tableaux CSV, des flux vidéo annotés ou des sous-titres au format ASS.
3. Traitement optimisé pour l'analyse à grande échelle, permettant des performances élevées lors de campagnes de surveillance médiatique.
4. Réduction des biais liés au genre, à l'âge et à l'origine ethnique grâce à l'utilisation de modèles d'apprentissage approfondi entraînés sur des jeux de données diversifiés.

### **5.3 speaking-clock-detection**

#### **5.3.1 Language**

Elle est développée en python.

#### **5.3.2 Pré-requis**

Ne fonctionne que sur des fichiers stéréo.

#### **5.3.3 Fonctionnalités Proposées**

Cet outil peut être utilisé pour détecter sur quel canal une horloge parlante est présente. En sortie trois valeurs sont possibles :

- **SPEAKING\_CLOCK\_TRACK** : suivi par l'identifiant du canal (0 ou 1).
- **SPEAKING\_CLOCK\_NONE** : aucune horloge parlante détectée.
- **SPEAKING\_CLOCK\_MULTIPLE** : plusieurs horloge parlante détectée ce qui est considérée comme une erreur.

## **6 Méthodes**

### **6.1 Transcription d'un audio en texte**

Cette méthode sera utile dans le cas où aucun métadonnées (tel que des sous titres) ne sont associés à la vidéo.

#### **6.1.1 Alternative 1 : SpeechRecognition**

Moteur de reconnaissance vocale : Google Web Speech API.

Seulement le format **WAV** pris en charge. Il est capable de transcrire une multitude de langue.

Temps de d'exécution de 37 secondes pour une vidéo de 2 minutes et 7 secondes, avec une très bonne précision plus de 95% de bonne correspondance sachant que le test a été effectué avec un anglais qui parle français.

Lien vers la vidéo dont j'ai extrais le contenu audio et convertis en .wav : Youtube - Le Parisien "le discours en français du Roi Charles III à Versailles"

Texte transcrit : *il nous incombe à tous de revigorer notre amitié pour qu'elle soit à la hauteur des défis de ce 21e siècle monsieur le Président Madame Macron je ne serai pour dire à quel point mon épouse et moi-même sommes ravi d'être parmi vous ce soir au thème de la première journée de notre première visite d'État en France et*

*combien nous sommes touchés par la magnifique accueil qui nous a été réservé une fois de plus la France et le peuple français nous ont témoigné un accueil chaleureux et une profonde gentillesse et nous leur ensemble très reconnaissant généralité nous rappelle que ma famille et moi-même avons été très ému par les hommages vendus en France à ma mère fait elle dans les funérailles ont eu lieu il y a une année je voudrais donc si vous ne le permettez porter un toast à monsieur le Président Madame Macron et le peuple français ainsi qu'à notre entente cordiale une entente pour le développement durable Christelle fidèle et constante à travers les siècles qui nous attendent contre vents et marée*

### 6.1.2 Alternative 2 : Google Cloud Speech-to-Text

Nécessite de créer un projet Google Cloud et en plus c'est payant (0.24€ par minute), donc je n'ai pas pu le tester.

Permet d'avoir une spatialité temporelle des phrases.

## 6.2 Gestion et extraction d'informations intégrées dans les fichiers vidéo

### Modes de stockage des sous-titres

Les sous-titres peuvent directement être gravés sur la vidéo donc on ne peut pas les extraire en tant que données car ils font partie de la vidéo elle-même.

Sous-titres multiplexés, dans ce cas ils sont inclus en tant que piste distincte dans le conteneur vidéo (le plus souvent dans des formats comme MKV, MP4 ou AVI). Ces sous-titres sont souvent au format SRT, ASS ou SUB.

Enfin les sous-titres peuvent être stockés séparément dans les mêmes formats que cité précédemment. Avec seulement du texte synchronisé avec la vidéo.

### Formats de Sous-Titres

SRT (SubRip Subtitle) : Texte brut avec des timestamps. Simple et largement utilisé.

ASS (Advanced SubStation Alpha) : Format plus avancé, permettant des styles, des couleurs et des animations.

SUB/IDX : Images de sous-titres (bitmap) avec un fichier d'indexation.

### Extraction

VLC comporte un système capable d'extraire les sous-titres et les exporter dans un fichier à part, mais il reste quand même assez limité dans sa prise en charge.

FFmpeg : outil qui permet d'une part d'afficher les métadonnées contenu dans le fichier vidéo (Principalement MKV et MP4).

```
Stream #0:0: Video: h264  
Stream #0:1: Audio: aac  
Stream #0:2: Subtitle: subrip
```

Figure 8: exemple de sortie

Mais aussi d'extraire des sous-titres et de le stocker dans un '.srt'.

### 6.3 Reconnaissance de texte dans les vidéos : Exploitation d'OCR (Reconnaissance Optique de Caractères)

Le moteur d'OCR **Tesseract** qui est développé par Google. C'est un programme autonome écrit en C++ et il existe *pytesseract* qui est un wrapper Python qui permet de l'utiliser dans un script python. Il est rapide mais pas toujours précis.

**EasyOCR** est une bibliothèque OCR basée sur des CNN et des Transformers développée par JaidedAI. Il supporte plus de 80 langues et il est beaucoup plus précis que Tesseract.

## Action sociale

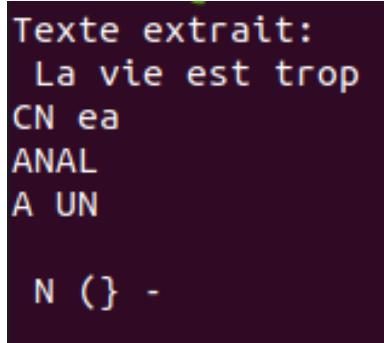
Le Département du Val-de-Marne mène une politique d'accueil et de soutien auprès de ceux qui rencontrent des difficultés sociales. Il assure la mise en place et le financement de dispositifs pour favoriser la lutte contre l'exclusion et réduire les inégalités.

Figure 9: premier test effectué



Figure 10: image où le texte est plus dur à identifier

Pour le premier test les deux moteurs sont capable d'extraire le texte de façon correcte. Pour la deuxième image voici le résultat obtenu pour Tesseract :



Texte extrait:  
La vie est trop  
CN ea  
ANAL  
A UN  
  
N {} -

Figure 11: texte extrait par Tesseract sur l'image 2

Le résultat pour EasyOCR est bien meilleur :

[**'La vie est'**, **'courte pour être'**, **'autre chose'**, **'quheureux'**, **'trop'**]

Sur la dernière image Tesseract n'extractit aucun texte tandis que EasyOCR arrive à extraire ce texte :

[**'SABINES'**, **'Montpellicr'**, **'Tomeration'**]

Il existe aussi MMOCR qui est utilise du Deep Learning et qui est censé avoir une meilleure précision mais que je n'ai pas testé.

## 6.4 Segmentations de scènes

### 6.4.1 Shot Boundary Detection (SBD)

Différents types de transition

Cut Transition

Fade out

Fade in

Dissolve

**Wipe Transition** Motion from bottom to top, Motion from right to left, Wipe with oblique motion, The movement is going to the center.

## 6.5 YOLO : You Only Look Once

La vitesse et la précision de YOLO en font un choix judicieux pour les tâches de détection d'objets en temps réel dans diverses applications.

### 6.5.1 Détection d'objets et identification

On peut l'utiliser pour détecter différent objets dans l'image. En chargeant des poids pré-entraînés on peut aussi identifier un certains nombres d'objets.

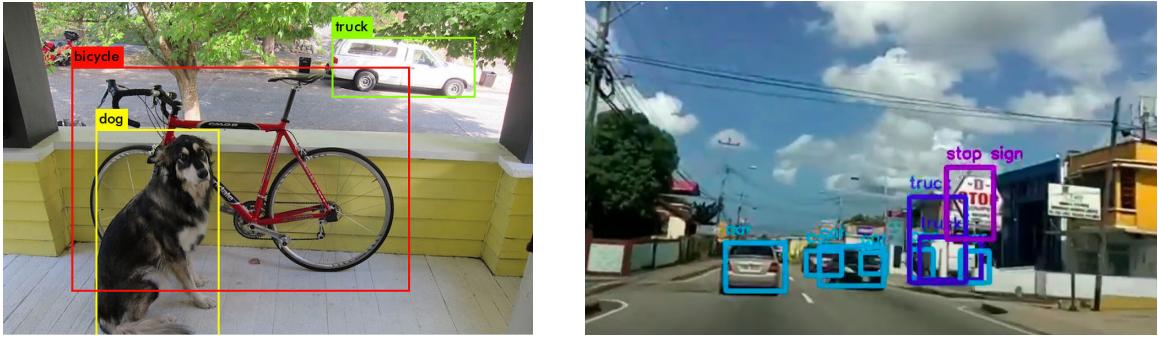


Figure 12: exemple d'utilisation

### 6.5.2 Poursuite de cible

De la même manière que précédemment pour la détection on peut aussi s'en servir pour suivre une ou plusieurs cible dans des images ou une vidéo.

## 6.6 Détection de Visage

**DeepFace** est un système de reconnaissance faciale basé sur l'apprentissage profond créé par un groupe de recherche de Facebook. Il identifie les visages humains dans les images numériques . Le programme utilise un réseau neuronal à neuf couches avec plus de 120 millions de poids de connexion et a été formé sur quatre millions d'images téléchargées par les utilisateurs de Facebook.

Cette librairie python propose différentes fonctionnalités qui sont les suivantes :

**Face Verification** A partir de deux images DeepFace est capable de terminer si oui ou non la personne de l'image 1 et bien celle de l'image 2 en extrayant de chaque image le visage. De base il utilisera la distance cosinus pour comparer les deux vecteurs caractéristiques extrait de chaque image respectivement, mais on a aussi la possibilité de faire la comparaison avec la distance euclidienne ou bien la distance euclidienne normalisé.

**Face Recognition** De la même manière que précédemment il est capable d'identifier si la ou les personnes présente dans une image appartient à la base de donnée.

**Embeddings** On peut aussi calculer le vecteur caractéristiques d'une image. La taille de ce vecteur peut varier en fonction du modèle utilisé, par exemple pour le modèle VGG-Face sa taille est de 4096.

**Face recognition models** DeepFace propose différents modèles, voici les plus populaires avec leurs précisions (calculé et non celle déclaré par les développeurs):

- VGG-Face (celui qui est utilisé de base) : Précision → 96.7%
- Google FaceNet : Précision → 97.4%
- OpenFace : Précision → 78.7%
- Facebook DeepFace : Précision → 69.0%
- DeepID : Précision → 66.5%
- Dlib : Précision → 96.8%
- ArcFace : Précision → 96.7%

**Facial Attribute Analysis** DeepFace a aussi un puissant module d'analyse des attributs du visage comprenant **l'âge**, **le sexe**, **l'expression faciale** (avec ces différentes expression faciale possible : la colère, la peur, le neutre, la tristesse, le dégoût, la joie et la surprise) et **la race** (asiatique, blanc, moyen-oriental, indien, latino et noir)



Figure 13: exemple de résultats pour l'analyse du visage

## Face Detection and Alignment

**Alignement** Il permet de garantir que toutes les images de visages sont dans une position cohérente, ce qui facilite des tâches comme la reconnaissance faciale, l'extraction de caractéristiques, ou la classification. Il est donc possible de choisir si on applique l'alignement ou non.

**Détection de visage** Il existe aussi différentes variantes de détection de visage :

- OpenCV (celui qui est utilisé de base) : c'est le détecteur de visage le plus léger. Il utilise un algorithme haar-cascade qui n'est pas basé sur des techniques d'apprentissage en profondeur. C'est pourquoi il est rapide, mais ses performances sont relativement faibles
- Dlib : Ce détecteur utilise un algorithme HOG, comme OpenCV il n'est pas basé sur l'apprentissage profond. Néanmoins, il présente des scores de détection et d'alignement relativement élevés.
- SSD : SSD signifie Single-Shot Detector ; il s'agit d'un détecteur populaire basé sur l'apprentissage profond. Les performances du SSD sont comparables à celles d'OpenCV. Cependant, le SSD ne prend pas en charge les repères faciaux et dépend du module de détection oculaire d'OpenCV pour s'aligner. Même si ses performances de détection sont élevées, le score d'alignement n'est que moyen.
- MTCNN : Il s'agit d'un détecteur de visage basé sur l'apprentissage profond et livré avec des repères faciaux. C'est la raison pour laquelle les scores de détection et d'alignement sont élevés pour MTCNN. Cependant, il est plus lent qu'OpenCV, SSD et Dlib.

- RetinaFace : RetinaFace est reconnu comme étant le modèle de pointe basé sur l'apprentissage profond pour la détection des visages. Cependant, cela nécessite une puissance de calcul élevée. C'est pourquoi RetinaFace est le détecteur de visage le plus lent par rapport aux autres.

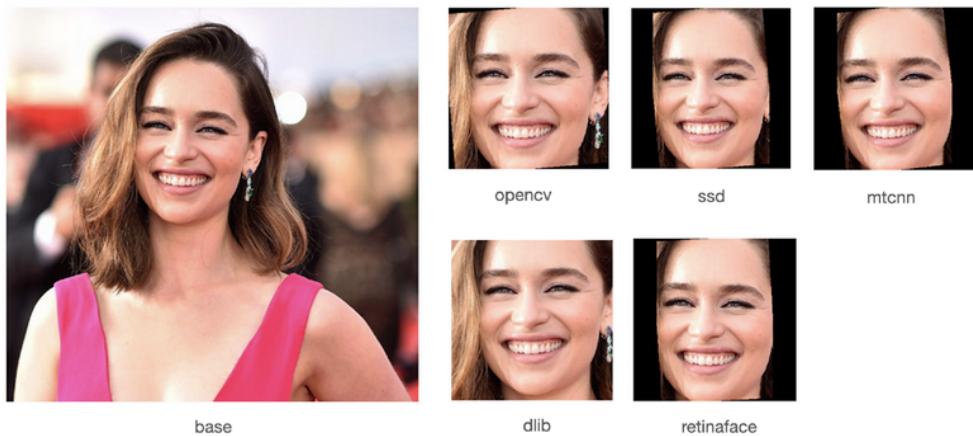


Figure 14: exemple de résultats pour la détection du visage

## 6.7 Optimisation des Performances

## 7 Définition

### 7.1 Étude iconographique

Étude descriptive des différentes représentations figurées d'un même sujet ; ensemble classé des images correspondantes.

### 7.2 Profilmique

Tout ce qui existe dans la réalité et qui pourra être enregistré par la caméra pour les besoins du film.

## 8 Références

- 3.1VLC Media Player - Page Wikipédia
- 3.2QuickTime - Page Wikipédia
- 3.3Screen - Site Officiel
- 3.4IINA - Site Officiel
- 3.5MPV - AlternativeTo
- 3.6DaVinci Resolve - AlternativeTo
- Comparaison des différents lecteurs
- 4.1Ronfard R. & Valero J., "KinoAI et le carnet audiovisuel : une solution plurielle pour l'étude des répétitions", European Journal of Theatre and Performance, mai 2020.

- 4.1KinoAI
- 5.1inaSpeechSegmenter - Github
- 5.2inaFaceAnalyser - Github
- 5.3speaking-clock-detection - Github
- 6.4.1Benoughidene A. & Titouna F., Shot Boundary Detection: Fundamental Concepts and Survey
- 6.6DeepFace - Documentation
- 6.6DeepFace - viso.ai