



CATIE



RÉGION
**Nouvelle-
Aquitaine**



UNION EUROPEENNE
Fonds Européen de
développement Régional

Outils d'imagerie pour la robotique - Sujet Projet

TS341_2026 — Robotique & Apprentissage - TS341

Version — 2026-10-05

CATIE

Sébastien Delpéuch s.delpéuch@catie.fr
Clément Pinet c.pinet@catie.fr

CATIE

Table des matières

1	Projet final TS341 - Décodage d'acquisitions brutes	3
1.1	Contexte	3
1.2	Données fournies	3
1.2.1	Confidentialité des données	3
1.3	Phase 1 : décodage	3
1.4	Phase 2 : optimisation	3
1.4.1	Mesure de la performance	4
1.4.2	Interface imposée	4
1.5	Bonus : floutage	4
1.6	Organisation	4
1.6.1	Groupe	4
1.6.2	Jalons	4
1.7	Rendu final	4
1.7.1	Structure du dépôt obligatoire	4
1.7.2	Rapport de projet	5
1.7.3	Outils obligatoires	5
1.8	Critères d'évaluation	5

1 Projet final TS341 - Décodage d'acquisitions brutes

1.1 Contexte

Un client du CATIE fait de l'inspection visuelle automatisée. Son banc d'acquisition est équipé de nombreuses caméras qui produisent des fichiers bruts. Le format de ces fichiers n'est pas documenté.

Le client a besoin d'un logiciel qui transforme chaque acquisition en une archive d'images propre, exploitable par la suite de sa chaîne de traitement. Le volume à traiter chaque jour est important : ce logiciel doit être rapide.

Le but de ce projet pédagogique dans le cadre du module TS341 est de reproduire le travail réalisé par le CATIE.

1.2 Données fournies

Les données sont disponibles via le lien suivant : [À compléter : lien Dropbox].

Vous recevez des dossiers d'acquisition tels que le client les produit, accompagnés de leurs fichiers .eop. Aucune documentation n'est fournie : tout ce qu'il y a à savoir se trouve dans les fichiers eux-mêmes et dans leurs noms.

Important : les données sont volumineuses (plusieurs gigaoctets par acquisition). Ne les committez pas dans votre dépôt Git.

1.2.1 Confidentialité des données

Ces données appartiennent à un client du CATIE. Elles sont confidentielles et vous sont confiées uniquement pour ce projet.

Le client met ses données à disposition de manière désintéressée. Le code que vous produisez reste votre propriété : il n'est analysé que par le CATIE, à des fins pédagogiques, et le client n'y aura jamais accès.

- Ne les partagez avec personne en dehors de votre groupe et des enseignants
- Ne les déposez sur aucun autre service : dépôt Git, stockage en ligne, messagerie, outil en ligne (convertisseur, visionneuse...)
- Ne publiez aucune image extraite de ces données, ni pendant ni après le projet
- Votre dépôt GitHub doit être **privé**. Seuls quelques extraits peuvent y figurer, pour illustrer le rapport
- Supprimez les données de vos machines à la fin du projet

1.3 Phase 1 : décodage

Du 06/10 au 19/11 : vous pouvez commencer dès la composition des groupes, les deux séances projet encadrées ont lieu le 13/11 et le 17/11.

Comprendre, par vous-mêmes, ce que contiennent les fichiers bruts, et les transformer en une archive d'images propre. Le format de cette archive est libre, mais doit être justifié.

La démarche compte autant que le résultat :

- Formulez des hypothèses sur le format, et vérifiez-les une par une
- Gardez une trace de ce que vous avez essayé, de ce qui a marché et de ce qui a échoué
- Appuyez chaque conclusion sur une preuve (une mesure, une image, un calcul)

Les cours de Clément Pinet (acquisition, traitement et analyse d'images) vous donnent la théorie nécessaire.

1.4 Phase 2 : optimisation

Du 20/11 au 15/12

Le 20/11, la spécification exacte de l'archive attendue vous est donnée. Elle remplace le format que vous avez choisi en phase 1 : un groupe bloqué en phase 1 peut donc quand même avancer.

L'objectif devient de produire cette archive le plus vite possible.

1.4.1 Mesure de la performance

- La mesure est le temps de traitement d'une archive, c'est-à-dire d'un dossier d'acquisition
- Elle est faite sur un PC aux spécifications fixes : [À compléter : CPU, RAM, disque]
- Le jeu de mesure est celui qui vous est fourni : il n'y a pas de données cachées
- Une archive non conforme à la spécification n'est pas chronométrée

1.4.2 Interface imposée

Pour que la mesure soit identique pour tous les groupes, votre logiciel est lancé ainsi :

```
docker build -t ts341-<groupe> .
docker run --rm \
  -v <dossier_donnees>:/entree:ro \
  -v <dossier_sortie>:/sortie \
  ts341-<groupe> /entree/<dossier_acquisition> /sortie
```

Le conteneur traite le dossier d'acquisition passé en argument et écrit l'archive dans /sortie.

1.5 Bonus : floutage

Flouter les visages présents sur les images de l'archive.

1.6 Organisation

1.6.1 Groupes

- 3 à 4 personnes par groupe
- Au moins un-e étudiant-e de chaque filière : Informatique, Électronique, Cognitique (ENSC)
- Composition à envoyer à s.delpeuch@catie.fr **avant le mardi 06/10/2026**, plus de changement possible ensuite

1.6.2 Jalons

Date	Jalon
mar. 06/10	Composition des groupes envoyée par mail
06/10 au 19/11	Phase 1 : décodage (2 séances projet)
ven. 20/11	Spécification de l'archive donnée à tous
20/11 au 14/12	Phase 2 : optimisation (4 séances projet)
mar. 15/12 23h59	Rendu final : code et rapport dans le dépôt

Il n'y a pas de rendu intermédiaire.

1.7 Rendu final

Date de rendu : 15/12/2026 23h59

Le dépôt GitHub sera automatiquement cloné à la date et heure du rendu. Il doit être privé, accessible et partagé avec sedelpeuch et clementpinet.

1.7.1 Structure du dépôt obligatoire

Vous devez réutiliser la structure du projet construite pendant les TP (fil rouge). Votre dépôt doit contenir :

1. Un package Python géré avec **uv** (`pyproject.toml` et `uv.lock` commités)
2. Un **Dockerfile** fonctionnel, qui respecte l'interface imposée ci-dessus
3. Un **README.md** : installation, utilisation, description des sorties

4. Un **rapport.md** (voir ci-dessous)
5. Des **tests** (pytest) et une **intégration continue** (GitHub Actions) qui les lance, avec le lint

1.7.2 Rapport de projet

Un fichier `rapport.md` à la racine du dépôt, court, qui couvre ces six points :

1. **Besoin** : reformulez le besoin du client avec vos mots
2. **Contraintes** : performances, matériel, formats, délais
3. **Découpage** : les grandes briques et leurs interfaces
4. **Choix techniques** : langages, bibliothèques, outils
5. **Risques** : ce qui a bloqué, et comment vous l'avez contourné
6. **Répartition** : qui a fait quoi, et quand

Pour la phase 1, le rapport décrit aussi votre démarche de décodage : hypothèses, vérifications, conclusions.

Chaque choix est comparé à au moins une alternative, avec un critère explicite. « OpenCV plutôt que scikit-image, car déjà vu en cours et plus rapide sur nos mesures » vaut mieux que « on utilise OpenCV ».

1.7.3 Outils obligatoires

- **uv** : gestion des dépendances et packaging
- **Typage** : annotations de types dans tout le code
- **Logging** : module `logging` de Python, pas de `print`
- **ruff** : lint et formatage
- **pytest** : tests
- **GitHub Actions** : intégration continue
- **Docker** : conteneurisation

1.8 Critères d'évaluation

Le projet est noté sur 20 points :

Critère	Points	Ce qui est évalué
Participation	5	En cours comme en séance projet
Phase 1 : décodage	5	Compréhension du format, démarche, archive produite
Phase 2 : optimisation	5	Conformité à la spécification, temps de traitement par archive
Qualité du code	5	Typage, logging, tests, lint, CI, conteneur