Università degli Studi di Modena e Reggio Emilia

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Laurea Magistrale in Ingegneria Informatica
a.a. 2019-2020

AImage Lab

Vision and Cognitive Systems

Final Project instructions

Rita Cucchiara

Project

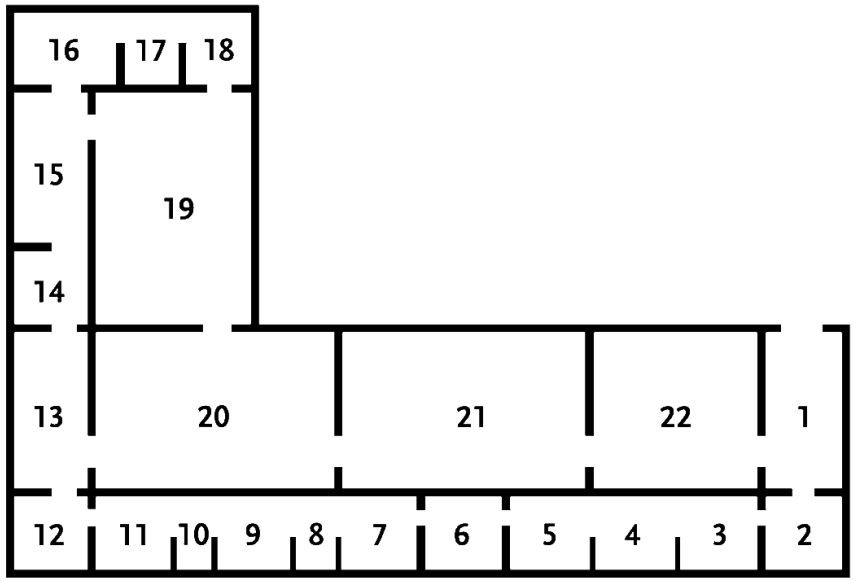
UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

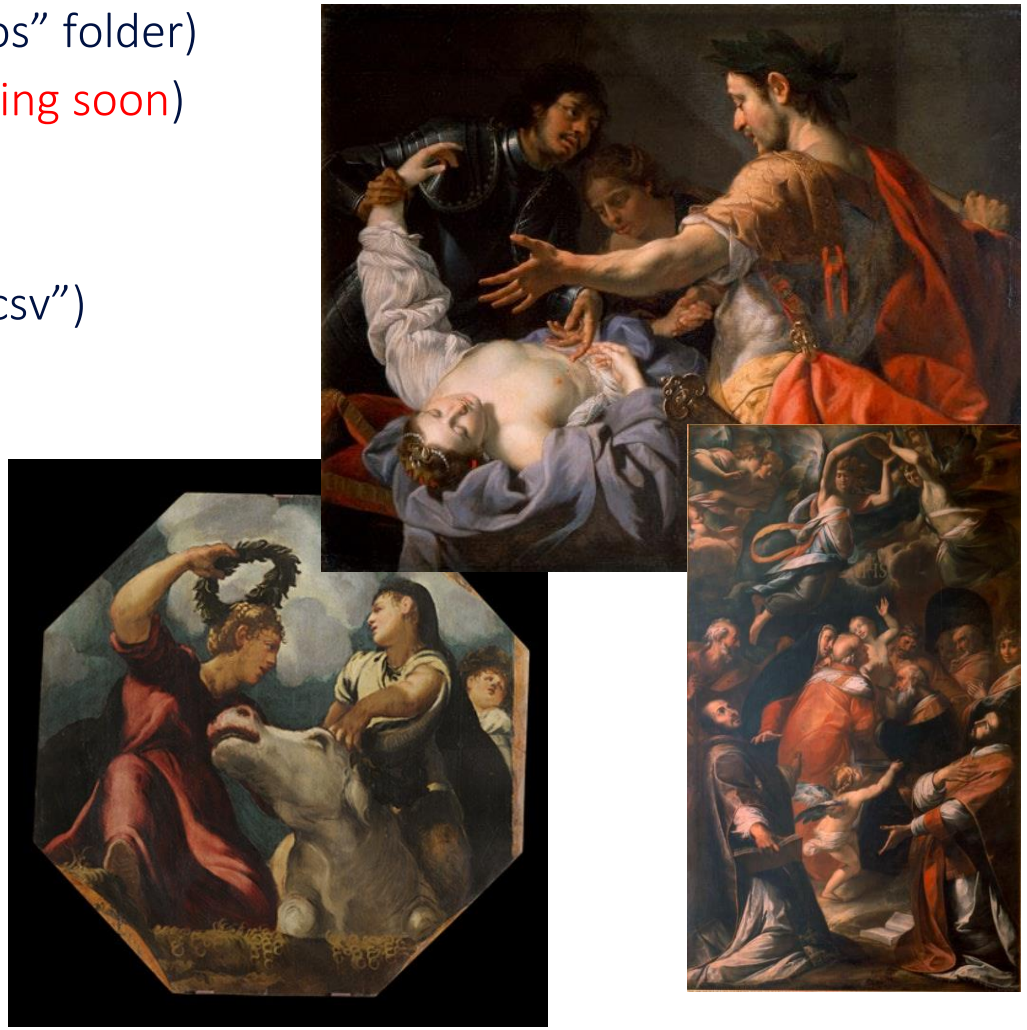
- You will be given videos and a 3D model taken in the same museum environment, plus:
  a. A map of the museum
  b. Pictures of all paintings (“paintings DB”)
  c. Mapping between paintings and rooms in the map

- Mandatory tasks:
  a. Painting detection: predict a ROI for each painting
  b. Painting rectification: correct the perspective distortion of each painting
  c. Painting retrieval: match each detected painting to the paintings DB
  d. People detection: predict a ROI around each person
  e. People localization: assign each person to one room on the map
- Optional tasks (complex):
  a. Determine whether each person is facing a painting
  b. Determine the distance between people and the door
  c. Replace paintings areas in the 3D model with the provided pictures

Data

- Inside the shared folder, you will find:
  - **Videos** of the museum, including some with people (“videos” folder)
  - A **3D model** taken in the same museum environment (coming soon)
  - A **map** of the museum (“map.png”)
  - **Pictures** of all paintings (“paintings_db” folder)
  - **Mapping** between paintings and rooms in the map (“data.csv”)





UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

AImage Lab

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

AImage Lab

Mandatory Tasks

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

- **Painting detection**: predict a ROI for each painting
  - Given an input video, your code should output a list of bounding boxes (x, y, w, h), being (x,y) the upper-left corner, each containing one painting.
  - create an interface to visualize given an image the ROI of a painting.
  - select painting and discard other artifacts.
  - (optional: segment precisely paintings with frames and also statues)
- **Painting rectification**: correct the perspective distortion of each painting
  - Given an input video and your detections (from the previous point), your code should output a new image for each painting, containing the rectified version of the painting.
  - be careful on not squared painting
- **Painting retrieval**: match each detected painting to the paintings DB
  - Given one rectified painting (from the previous point), your code should return a ranked list of all the images in the painting DB, sorted by descending similarity with the detected painting. Ideally, the first retrieved item should be the picture of the detected painting.
- **People detection**: predict a ROI around each person
  - Given an input video, your code should output a list of bounding boxes (x, y, w, h), being (x,y) the upper-left corner, each containing one person.
- **People localization**: assign each person to one room on the map
  - Given an input video and people bounding boxes (from the previous point), your code should assign each person to one of the rooms of the Gallery. To do that, you can exploit the painting retrieval procedure (third point), and the mapping between paintings and rooms (in “data.csv”). Also, a map of the Gallery is available (“map.png”) for a better visualization.

Optional tasks

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

- Determine whether each person is facing a painting
  - Given an input video, people and paintings' detections, determine whether each person is facing a painting or not.

- Replace paintings areas in the 3D model with the provided pictures
  - Given a view taken from the 3D model, detect each painting and replace it with its corresponding picture in the paintings DB, appropriately deformed to match the 3D view.

- Determine the distance of a person to the closest door
  - Find the door, find the walls and the floor, try to compensate and predict distance

Concluding remarks

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

- You are free to organize your code as you like, as long as it produces all the mandatory outputs.
- You are free to use any library of your choice, and any pre-trained or custom network. Try to use algorithms learned during the course and discuss their limits or possible alternatives.

- You will need to upload a single .zip file, containing:
  - All your code, under the “code” folder
    - This might be either Python source files, IPython notebook files, or a combination of the two.
    - The source code should clearly indicate how to reproduce the proposed pipeline, and where the required outputs are computed
    - Additionally, any external resource required to execute the code, apart from the dataset and freely available libraries, should be included in the package.
    - If any external packages are used, you are kindly invited to provide either a readme file or a requirements.txt file.
  - A report, in a single PDF file named “report.pdf”.
    - A sample (non mandatory) template is available.
    - The report should include title, authors, introduction, a related works section, a section on the approach, a results section, discussion and bibliography.

Deadline

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

- 10 June 2020 for people that aim to attend final exam in summer (June - July)

- 10 September 2020 for people that aim to attend final exam in fall/winter (September - November)