

Leiden University
4032COVIX: Computer Vision
Autumn 2025.
Instructors: Hazel Doughty, Lu Cao

Assignment 3: Gesture Recognition

1 Goal

The goal of this assignment is to build and train a model to recognize hand gestures in video using the Jester dataset.

You will design, implement, and evaluate a gesture recognition pipeline that demonstrates your understanding of computer vision techniques and your ability to reason about model design, evaluation and limitations.

You can work individually or **in pairs**.

2 Data

The dataset used in this assignment is the Jester dataset, which contains short video clips showing 27 different human hand gestures.

The full training set contains 118,562 videos, and the validation set contains 14,787 videos. Since the labels from the test set are not publicly available, you will evaluate your models on the validation set.

To make experimentation more accessible, we provide a zip file containing a 20% subset of the training data along with the complete validation set.

You are free to use as much of the original training data as you wish, but you should always evaluate your final model using the entire validation set. If you use a subset of the training data for hyperparameter tuning or early validation, describe this clearly in your report.

The full dataset can be downloaded from: <https://www.qualcomm.com/developer/software/jester-dataset/downloads>

3 Starter Code

We will not provide starter code for this assignment. You are expected to build all components of the pipeline yourself, including data loading and model training.

You are, however, free to reuse code from previous assignments, practical notebooks or publicly available repositories, provided that you clearly acknowledge any external code used in your report.

4 Rules

Unlike previous assignments, you are free to experiment with whatever method and/or data you wish to solve the problem. However, you must write a report about your project and your project must meet the following **core requirements**:

1. **Constraint Definition:** Define and justify one clear goal or constraint that guides your project. This could be for example:

- Overall accuracy, e.g. achieving the highest possible accuracy with the available compute
- Compute or training time limitation, e.g. building a good model that trains in under 2 hours on a lab GPU
- Data limitation, e.g. training a model with only 5000 videos
- Generalization, e.g. testing your model on newly collected or heavily augmented gestures to test the robustness of the model

State your chosen constraint and explain how this constraint influenced your design choices and results.

2. **Baseline and Improvement:** Implement at least one baseline model (e.g. a simple 2D CNN operating on a randomly sampled frame) and one improved version (e.g. using frame stacking, optical flow, temporal modelling or data augmentation). Compare their results and discuss the differences.

3. **Evaluation:** Your report must include both:

- **quantitative evaluation**, numerical results such as accuracy, per class accuracy or a confusion matrices that measure how well your model performs.

- **qualitative analysis** - visual or descriptive analysis that helps interpret why the model behaves as it does. Examples include showing correctly and incorrectly classified examples, visualizing activation maps (e.g., Grad-CAM), or describing observable failure patterns.

You will be graded on how well you motivate, implement, and analyse your approach, not on model accuracy.

5 Report

The report should be **4 pages** in length (excluding references). Points will be deducted for excessively long or short reports.

If working in a pair, you should write the report together but you must include a section that lists the individual contributions of each team member.

Structure your report like a short research paper, including:

- Abstract
- Introduction and Related Work (including description of chosen constraint)
- Methodology (including your baseline and improvement)
- Experiment Results and Discussion
- Conclusion
- References

In your discussion, analyze both strengths and weaknesses of your approach, including the effects of your design choices, your constraint, and any failures or limitations you observed.

We recommend you write the report in the CVPR format.

6 Submission

Submit your report to BrightSpace in **PDF** format alongside a zip file of the code.

If working in a pair, only one submission is required, but both student names and IDs must appear at the top of the document.

7 Grading

Your grade depends on how well you explain what you did, why you did it and how you obtain and analyze the results. Specifically, we will grade according to the following criteria:

1. Problem, Data and Related Work description (15%)
2. Implementation description (25%)
3. Application of Computer Vision Techniques (20%)
4. Results and Discussion (30%)
5. Structure (5%)
6. Readability (5%)

You can find a description of the grading criteria in the other pdf attached to the assignment.

8 Running your code on the lab machines

It may be useful to run your code on the lab machines in DM.0.09, etc. This is relatively easy to set up. First create a conda environment with:

```
conda create --name pytorch python=3.9
```

activate it with

```
conda activate pytorch
```

and install pytorch with

```
conda install pytorch torchvision torchaudio pytorch-cuda=12.4 -c pytorch -c nvidia
```

You can also access these machines remotely. Instructions are attached to the assignment.