Leiden University
4032COVISY: Computer Vision
Autumn 2024. Instructor: Hazel Doughty

Assignment 3: Gesture Recognition

1 Goal

The goal of this assignment is to build and train a model to recognize hand gestures in video. For this assignment you can **work in pairs**.

2 Data

The dataset we'll use is the Jester dataset, consisting of video clips of 27 different human hand gestures. The training set of the Mini Places dataset has 118,562 videos; the validation set has 14,787. Since the labels from the test set are not released you will evaluate your models on the validation set. If you want to use a validation set to choose hyperparameters for your model we suggest you create your own subset for this from the training data. The data can be downloaded from here: `https://www.qualcomm.com/developer/software/jester-dataset/downloads`

3 Starter Code

We will not provide starter code for this assignment. You have to build each aspect necessary including data loading but you are free to reuse code from previous assignments, the practical notebooks and other code found online.

4 Rules

Unlike previous assignments there are no rules, you are free to experiment with whatever method you wish to solve the problem. You will also be assessed via a written report rather than by the accuracy of your model. This means you do not necessarily need to aim for a highly accurate model, you can instead aim for a fast model to limit the compute resources needed.

5 Report

The report should be 4 pages in length (excluding references). We will deduct points for excessively and unnecessarily long reports (a well-written concise report is better than a long and wordy one!). If you're working in a pair, the expectation is that you'll likely need the full 4 pages for your report. Furthermore, if you're part of a pair, you should write the report together but you must include a section that lists the individual contributions of each team member. We will apply a penalty if this section is missing. The report should be structured like a research paper, starting with an abstract, followed by sections for Introduction, Related Work, Methodology, Experimental Results, Discussion, Conclusion, and ending with references. Some of the sections can be combined if you want (specifically, Introduction/Motivation & Related Work as well as Results & Discussion). We recommend you write the report in the CVPR format.

You should describe and evaluate what you did in your project, which may not necessarily be what you hoped to do originally. A small result described and evaluated well will earn more credit than an ambitious result where no aspect was done well. Be accurate in describing the problem you tried to solve. Explain in detail your approach, and specify any simplifications or assumptions you have made. Also demonstrate the limitations of your approach. When doesn't it work? Why? What steps would you have taken had you continued working on it? Make sure to add references to all related work you reviewed or used.

6 Submission

Submit your report to BrightSpace in PDF format. If you're part of a team, only one of you should submit the PDF. In that case, please list both partners' names and student ID numbers at the top of the PDF.

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 (20%)
3. Application of Computer Vision Techniques (20%)
4. Results and Discussion (20%)
5. Structure (10%)

6. Readability (15%)

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
```