

INF6804 Computer Vision

H2026 – Practical Assignment #1

Sports Action Recognition via Feature Embeddings

Objectives:

- Learn to perform action recognition using a feature embedding approach, without end-to-end training.
- Implement and compare a classic feature extractor with a deep learning-based extractor.
- Implement and analyze two simple classification strategies: nearest centroid and k-NN.
- Analyze the trade-offs between methods in terms of accuracy, speed, and robustness.
- Formulate hypotheses and critically evaluate them based on experimental results.

Submission:

- Submit before **the 6th of February, 17:00** –*late submissions will not be accepted*
- **Where to submit:**
 - *Source Code:* Submit your code on **Moodle** (we should be able to run your tests).
 - *Report:* Submit your report (*.pdf format*, 8 to 15 pages, font size 10) on **Gradescope**.

References:

- See course notes on Moodle (Chapter 1)

Other directives:

- The assignments must be made in teams of two, submit only one version of your work!

Presentation

In this assignment, you will build and evaluate a system for sports action recognition. Instead of training a deep network from scratch, you will use a feature embedding approach. You have to compare between two methods and determine which one is better at our task, which is finding the accurate sport label for a video. The first method is based on **Histogram of Oriented Gradients (HOG)**, the second is based on **Pre-trained Convolution Neural Network (ResNet)**. You can use your course notes as a reference to understand their basic working principles, and you can look online for more details.

To compare the two methods, you will have to evaluate how accurate the model is at classifying the video in the right category, the workflow is as follows: for a set of reference videos, you will extract features from individual frames to build a representation for each sports class. Then, for a new test video, you will extract its features and classify it based on its similarity to the reference class representations. In order to classify the video, we compare between two classification strategies:

- **Mean Vector (Centroid):** Represent each class by the average feature vector of all its reference frames.
- **k-Nearest Neighbors (k-NN):** Store all reference features and classify a test video's frames by a majority vote of their nearest neighbors.

Using the ("*Push ups*", "*Handstand Pushups*", "*Biking*", "*Playing Cello*", "*Playing Violin*") classes of the **UCF101 dataset**, you must include the following in your report (marked on 20 pts):

Report Structure and Questions

1. Presentation of Methods (2 pts):

In your own words, describe the core principles of the two feature extractors you are comparing:

- How does **HOG** work? What kind of information does it primarily encode (e.g., texture, shape, color)?
- How does a **ResNet** work? What kind of features do you expect from its final layers?
- Briefly explain the **Nearest Centroid** and **k-NN** classifiers. What is the main difference?

2. Performance Hypotheses (2 pts):

Based on your theoretical understanding, formulate hypotheses about the expected performance:

- **Extractor Comparison:** Which feature extractor (HOG or ResNet) do you predict will achieve higher classification accuracy on the given sports classes? For example, which model will be better at detecting the class "Biking".
- **Classifier Comparison:** For a given feature type (e.g., ResNet features), which classification strategy (Centroid or k-NN) do you expect to perform better? Explain the potential trade-offs in terms of accuracy, memory usage, and prediction speed.

3. Hypotheses on Robustness (2 pts):

In this question, you "attack" your best model by augmenting the test videos. Before you run the experiment, hypothesize:

- Which feature extractor (HOG or ResNet) do you believe will be more robust to **lighting changes** (e.g., a large shift in brightness)? Why?
- Which do you believe will be more robust to a **geometric change** like a 90-degree rotation? Why?

4. Description of Experiments and Implementation (3 pts):

Describe your experimental setup:

- How did you sample frames from the videos to create your reference and test sets?
- Describe the implementation. Which libraries did you use (e.g., OpenCV, Scikit-learn, Hugging Face)? What were the key parameters for HOG and ResNet (e.g., cell size for HOG)? How did you set the value of k for k-NN?

- Which evaluation metrics did you use to measure performance (e.g., overall accuracy, confusion matrix)? Justify your choice.

5. Experimentation Results (4 pts):

Present your results clearly using tables and figures.

- Provide a table comparing the classification accuracy of all four combinations (HOG+Centroid, HOG+kNN, ResNet+Centroid, ResNet+kNN).
- Show the confusion matrix for your single best-performing model combination.
- Provide a table or graph showing the results of your "robustness attack" experiments from Question 3 (performance on original vs. augmented test data).

6. Discussion and Analysis (4 pts):

Analyze your results in depth:

- Discuss your results from Question 5 in relation to your hypotheses from Questions 2 and 3. Were your predictions correct? If not, why might the results have differed from your expectations?
- Which method (HOG or ResNet) proved superior? Analyze the confusion matrix of your best model: which classes were easily distinguished, and which were confused? Why might this be the case?
- What are the final, measured trade-offs (accuracy, feature extraction time, classification time) between the different approaches? Which combination would you recommend for a real-world application and why?

7. Readability and completeness (3 pts):

In addition to the content, the format must be well-structured and complete.

Resources

- **Dataset:** UCF101: <https://www.crcv.ucf.edu/research/data-sets/ucf101/>.
- **Classic Descriptors:** OpenCV, scikit-image.
- **Deep Learning Frameworks:** PyTorch, Hugging Face Transformers, TensorFlow.