

Machine learning for vision and multimedia

November 2025

1 Project Organization and Enrollment

Projects must be carried out by **groups of students**; each group consists of **1–3 members**. Based on previous years' experience, **larger groups** allow a better distribution on workload. Project capacity is **limited**: as reservations are submitted, availability decreases, and groups may enroll only in projects that still have available slots.

1.1 Group Formation and Project Proposal Submission

- All students must log in to Moodle platform
- A group must be created for each subset of students on the platform via **Groups Selection**. (*available from 2025-11-13 00:00*)
- The group representative must select the project via **Project Choice** on Moodle; **only one member needs to do this**, and the selection will automatically apply to all the members of that group. Each project can be chosen by a **limited number of groups**; if it is no longer possible to select a project, it means all available slots for that project have been filled. (*available from 2025-11-19*)
- Each group must submit a **Project Proposal**. The proposal, **to be submitted before** starting the development activities, must include, for the chosen project: (*available from 2025-11-19*)
 - the dataset intended for use (with source);
 - the architecture to be adopted;
 - the training setup;
 - the evaluation metrics to assess performance.

1.2 Scope, Common Guidelines, and Deliverables (Applies to All Projects)

Documentation, Report, and Submission

Each group must maintain comprehensive documentation of design, training, and evaluation, including preprocessing; the train/validation/test split with percentages and the underlying rationale; validation criteria and decision thresholds; loss functions and optimization strategies; quantitative and qualitative results; and an error/confusion analysis with a discussion of failure cases. The final report (6–8 pages) must present the application context and operational definitions, the dataset with statistics, the adopted machine learning techniques and hyper-parameters, the training/validation protocols, the performance metrics, the results, and a balanced discussion of limitations and possible future developments. The submission package must include the full project code and/or notebooks, the weights of at least one trained model, scripts to regenerate all tables and figures, and (when the dataset is not publicly available) a copy of or a link to the dataset; it must also contain a README sufficient for full reproducibility (covering setup, training, and evaluation) and the compiled self-assessment checklist. Optionally, groups may also provide an experiment log (e.g., using the provided Sample ML tracker spreadsheet) and supplementary material such as additional figures, videos, or other artifacts that support the evaluation.

Assessment and Grading (Applies to All Projects)

Grading is based on the following pillars (building on the documentation and results required above, without repeating them):

- **Reproducibility & documentation:** completeness and clarity of code, README, seeds, and scripts.
- **Experimental rigor:** sound splits and protocols, well-motivated design choices, and ablation studies where appropriate.
- **Results & error analysis:** quality of quantitative and qualitative evaluation, including confusion/error analysis and discussion of failure cases.
- **Critical discussion:** insight into limitations, bias, and generalization, with well-argued future work.

Task-Specific Metrics

Use the appropriate subset of metrics below for the specific project type (provided here, non-exhaustively, for each project to avoid repetition).

- **Detection & Classification:** mAP@0.5:0.95, precision–recall curves, confusion matrix analysis, error analysis.

- **Semantic Segmentation:** mIoU, per-class IoU/Accuracy, boundary F1, attention to small regions and occlusions.
- **Anomaly Detection:** AUROC/AUPRC, threshold calibration; (optional) PRO for localization.
- **Logo Retrieval:** mAP, Precision@K, Recall@K.
- **Audio (SER, AMT, Denoising, RIR):** SER: macro-F1/accuracy; AMT: note/onset F1; Denoising: SI-SDR, STOI, PESQ; RIR: MAE/RMSE on T60/DRR.
- **System Identification:** MSE/NRMSE (one-step and multi-step), training time, number of parameters/FLOPs.
- **HRC Safety:** Accuracy/F1/ROC-AUC or MAE/RMSE for continuous risk; subject-wise splits.

1.3 Tutoring and Lab Sessions

For each project, a **Point of Contact (POC)** – a researcher from the group of the teaching staff – is indicated in the project description; students may contact the POC for clarifications specific to the project text.

For all other requests (organization, assessment, general guidance), please contact Dr. Rosario Milazzo (milazzo.rosario@polito.it) or Dr. Francesco Manigrasso (manigrasso.francesco@polito.it). For any Moodle-related requests or issues, please contact Dr. Francesco Manigrasso.

The three planned lab sessions will be entirely dedicated to the project, and participants will be divided into working groups organized around the activities scheduled for each session:

- **Lab 1 on Thursday, 04/12/2025** will focus on identifying the objective, defining the classes, and collecting or selecting the data;
- **Lab 2 on Thursday, 18/12/2025** will focus on implementing and training the reference model, documenting design choices and metrics;
- **Lab 3 on Thursday, 08/01/2026** will focus on continuing training and evaluation of the model, consolidating the pipeline and reporting results.

2 Projects for 2025/2026

2.1 Detection + Classification of State/Action on a Chosen Entity

Objective

Design and implement a computer vision system that, given images, locates an entity in the scene (object, person, or animal) and assigns it a label from a set

of n proposer-defined classes. The classes must be clearly distinct and accompanied by precise operational definitions to ensure consistency in annotation and interpretation.

Dataset

For each usable instance, provide: class label; coordinates of the region of interest (bounding box or another agreed representation); explicit data source; overall dataset statistics (counts per class, distribution characteristics, image resolution, domain variations).

Experimental Plan

- Phase 1: Train and evaluate a model on a baseline dataset.
- Phase 2: Propose and evaluate methods to mitigate class imbalance. In this phase, the training dataset must be intentionally unbalanced, meaning that some classes must be significantly underrepresented (for example, the smallest class has less than 10–20% of the samples of the largest class). A model must be trained with the newly created training dataset, introducing methodologies to address the problem. Each strategy must be clearly justified, with quantitatively demonstrated effectiveness compared to the Phase 1 baseline.

Example Datasets

COCO: <https://cocodataset.org/>, Pascal VOC: <https://www.robots.ox.ac.uk/~vgg/projects/pascal/VOC/>.

Point of Contact

Francesco Manigrasso (manigrasso.francesco@polito.it).

2.2 Detection of Anomalies with Possible Localization

Objective

Design and implement a computer vision system capable of detecting anomalies in a controlled visual domain. In this context, anomaly detection refers to the automatic identification of images or regions that deviate from the normal appearance learned from reference data. The system must therefore learn what is normal—based on images acquired under controlled conditions—and recognize visual deviations that indicate potential defects, irregularities, or unexpected elements. Consider both image-level (normal/anomalous classification) and, when possible, pixel-level (defect localization) analyses.

Dataset

Explicitly define what is “normal” and what constitutes an “anomaly.” Collect a large number of normal images and a limited number of anomalous ones; provide masks of defective regions for a subset of anomalous cases. May be collected ad hoc or derived from a public source; must include, for each instance, the image-level label, the data source, and—when available—region or pixel-level annotations (masks).

Experimental Plan

- Phase 1: Train exclusively on normal data under controlled conditions (e.g., homogeneous illumination, clean surfaces) to learn nominal appearance and calibrate decision thresholds.
- Phase 2: Evaluate robustness under domain shift (illumination, viewpoint, camera, background). Train and test on new normal and anomalous images acquired under these conditions; report impact on detection and localization. The dataset may be synthetically altered to emulate the target domain (e.g., photometric changes, sensor noise, geometric perturbations, background substitutions), but alterations must correspond to plausible real-world cases and be documented. Propose and evaluate techniques to mitigate observed degradation, with clear justification and quantitative evidence.

Example Datasets

MVTec AD: <https://www.mvtec.com/company/research/datasets/mvtec-ad>.

Point of Contact

Francesco Manigrasso (manigrasso.francesco@polito.it).

2.3 Semantic Segmentation

Objective

Semantic segmentation assigns each pixel a semantic label from a predefined set, producing a full-scene membership map. Design a system that, within a reference environment, selects n macro-classes and generates a per-pixel class map for each image or clip.

Dataset

May be collected ad hoc or drawn from a public source. Must include pixel-wise masks for a well-covered subset, with explicit indication of the data source.

Experimental Plan

- Phase 1: Train and evaluate a baseline model on a baseline dataset.
- Phase 2: Propose and evaluate methods to mitigate class imbalance. In this phase, the training dataset must be intentionally unbalanced or occlusions or controlled masking must be artificially introduced to mimic real-world challenges. Some classes must be significantly underrepresented (e.g., the smallest class has less than 10–20% of the samples of the largest class). Occlusions can be artificially introduced through augmentation or controlled masking to simulate real-world challenges. Analyze and mitigate the impact of these conditions and the identified class imbalance; evaluate whether the strategies lead to measurable improvements compared to the baseline in Phase 1.

Example Datasets

Pascal VOC Segmentation: <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/>.

Point of Contact

Francesco Manigrasso (manigrasso.francesco@polito.it).

2.4 Attribute Multi-Task: Category + Discrete Attribute + Continuous Measurement

Objective

Design and implement a multi-task vision system that, for each image/clip, produces three outputs: (i) a category (class), (ii) a discrete attribute (e.g., material or typology), (iii) a continuous measure in $[0, 1]$ (e.g., quality, degree of wear, normalized confidence).

Dataset

May be collected ad hoc or derived from a public source and must include consistent annotations for all three targets.

Experimental Plan

- Phase 1: Train a baseline model on a baseline dataset and evaluate performance across the three tasks, providing a cross-task consistency analysis to verify whether outputs are mutually consistent or contradictory.
- Phase 2: Include a compositional generalization experiment: maintain a subset of class-attribute pairs in training (these can also be artificially obtained via controlled synthesis/augmentation). Analyze performance

and task-to-task consistency on held-out combinations, highlighting generalization to unseen compositions (e.g., if training includes car-red and bike-blue, test on car-blue) and expose failure modes.

Example Datasets

CelebA (attributes + continuous scores): <https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>.

Point of Contact

Francesco Manigrasso (manigrasso.francesco@polito.it).

2.5 System Identification of Nonlinear Dynamical Systems

Objective

Perform system identification (i.e., learning the model of a dynamical system) of challenging nonlinear systems.

Dataset

Choose one dataset from the Nonlinear System Identification Benchmarks: Industrial Robot, Cortical Responses Evoked by Wrist Joint Manipulation, or F-16 Ground Vibration Test.

Experimental Plan

- Phase 1: Train different kinds of models: NNARX networks, simple recurrent neural networks (RNNs) and LSTM RNNs (optionally, GRU RNNs can also be considered).
- Phase 2: Explore different architectures (number of states, number of layers, etc.) for each kind of model.
- Phase 3: Compare the obtained results in terms of: model accuracy in simulation; time required to perform the training; model complexity: number of parameters and FLOPS to perform the output sample prediction.

Example Datasets

Nonlinear Benchmarks: <https://www.nonlinearbenchmark.org/benchmarks>.

Point of Contact

Lia Morra (lia.morra@polito.it).

2.6 Few-Shot Logo Recognition

Objective

Train a few-shot logo detector which, starting from one or more images of the logo to be recognized, is capable of retrieving all instances of the same logo from a dataset.

Dataset

May be collected ad hoc or derived from a public source.

Experimental Plan

- Phase 1: Starting from a dataset of annotated images, design a proper experimental plan (training, validation and test set, evaluation protocol and metrics) suitable for designing.
- Phase 2: Establish a baseline using pre-trained models as embeddings.
- Phase 3: Refine the above-mentioned model with a suitable learning strategy for logo re-identification.

Example Datasets

LogoDet-3K-Dataset: <https://github.com/Wangjing1551/LogoDet-3K-Dataset>.

Point of Contact

Lia Morra (lia.morra@polito.it).

2.7 Predictive Safety and Dynamic Risk Estimation in Human–Robot Collaboration

Objective

Design and implement a data-driven model for safety prediction in human–robot collaborative workcells equipped with 7-axis cobots. The goal is to estimate the safety state of an interaction (e.g., safe, warning, or critical) from multi-modal sensor data that describe both robot motion and human pose dynamics. Building upon recent work on human–robot trajectory forecasting in industrial settings, the project focuses on learning to recognize and anticipate potentially unsafe configurations from synchronized motion data, providing a predictive foundation for what could serve as a mechatronic safety layer in future collaborative robotic systems.

Dataset

Use a public dataset of human–robot collaboration that includes synchronized recordings of robot joint trajectories, end-effector motion, and human 3D pose or position, together with safety annotations (e.g., safe, near-collision, collision). Each instance should include time-aligned robot and human data. Dataset statistics (number of samples, subjects, events, and class balance) must be documented and analyzed.

Experimental Plan

- Phase 1: Train a classification or regression model (e.g., MLP, random forest, or CNN depending on data representation) to predict the safety level from single-frame or aggregated features.
- Phase 2: Extend the baseline with sequence models (e.g., LSTM, GRU, or 1D CNN) to capture short-term dynamics of human–robot interaction and predict future risk states.
- Phase 3: Perform quantitative and qualitative evaluation of model behavior, including feature importance or saliency analysis to identify which motion or proximity patterns contribute most to unsafe predictions.

Example Datasets

CHICO-PoseForecasting: <https://github.com/AlessioSam/CHICO-PoseForecasting>, LiHRA: <https://doi.org/10.5281/zenodo.16675029>.

Point of Contact

Davide Calandra (calandra.davide@polito.it).

2.8 Automatic Music Transcription

Objective

Design and implement a deep learning system capable of automatically transcribing monophonic (one note at a time) or polyphonic (multiple notes sounding simultaneously) music recordings of a single musical instrument into symbolic notation (e.g., MIDI). The model must learn to map audio waveforms or time-frequency representations (e.g., spectrogram, Mel, CQT) to sequences of pitch (note frequency) and onset (note start and duration) events. Consider single instrument recordings and focus on one instrument (e.g., piano, guitar, violin, flute, etc.).

Dataset

- Use a publicly available dataset of single-instrument performances that provides aligned audio and symbolic annotations. Each sample should include: audio recording (preferably isolated instrument stems); ground-truth pitch and onset annotations (e.g., MIDI note and event lists).
- Alternatively, create a synthetic dataset using software synthesizers such as FluidSynth or Virtual Instrument plugins for Digital Audio Workstations. Render audio from MIDI files. Document the synthesis configuration to ensure reproducibility.

Experimental Plan

- Phase 1: Train a baseline model (e.g., CNN) that predicts pitch of single or multiple notes from spectrograms.
- Phase 2: Extend the baseline model to retrieve both pitch and note onset/offset times, possibly via a multi-output architecture or post-processing step.
- Phase 3: Perform qualitative and quantitative analysis of the model behavior comparing different time-frequency representations (e.g., STFT, Mel, CQT). Evaluate model robustness to recording/synthesis variations (e.g., reverb, detuning, noise). Optionally, assess the contribution of temporal sequence models (RNN, LSTM) relative to pure convolutional baselines.

Example Datasets

MAESTRO: <https://magenta.tensorflow.org/datasets/maestro> MusicNet: <https://zenodo.org/records/5120004> Slakh: <http://www.slakh.com/> NSynth: <https://magenta.tensorflow.org/datasets/nsynth>.

Point of Contact

Antonio Servetti (antonio.servetti@polito.it).

2.9 Speech Emotion Recognition

Objective

Design and implement a machine learning system capable of recognizing the emotional state expressed in human speech. The goal is to map acoustic and prosodic features (e.g., pitch, energy, formants, spectral shape) to discrete emotion labels such as anger, happiness, sadness, fear, surprise, neutral, or others as defined by the group. The system may process short utterances or continuous speech, and may target speaker-dependent or speaker-independent recognition.

Dataset

Use a public dataset of emotional speech that includes audio recordings and corresponding emotion annotations. Each instance should include: audio file (speech utterance, sentence, or conversation segment); emotion label (categorical, e.g., happy, sad, angry, neutral) or continuous arousal/valence values; speaker metadata (gender, language, ID if available).

Experimental Plan

- Phase 1: Train and evaluate a baseline model (e.g., CNN, LSTM, or CRNN) on spectrogram or Mel-frequency cepstral coefficients (MFCC) representations of speech to predict categorical emotions. Document pre-processing (normalization, silence trimming, segmentation), network architecture, and validation metrics.
- Phase 2: Extend the model with the following strategies:
 - Perform data augmentation (e.g., noise addition, pitch/time shifts, reverberation) to test robustness under variable acoustic conditions.
 - Incorporate speaker-independent evaluation (train/test on disjoint speaker sets).
 - Each improvement must be justified and quantitatively compared to the baseline model from Phase 1.
- Quantitative: accuracy, F1-score, confusion matrices, and optionally regression metrics (MAE, RMSE) for continuous labels.
- Qualitative: Error analysis focusing on confusion between similar emotional states (e.g., happy vs surprised). Discussion of generalization limits, cross-speaker variability, and dataset bias.

Example Datasets

SER-datasets collection: <https://github.com/SuperKogito/SER-datasets>.

Point of Contact

Antonio Servetti (antonio.servetti@polito.it).

2.10 Generative Adversarial Networks for Data Augmentation and Domain Adaptation

Objective

Design and implement a generative adversarial network (GAN)-based system to augment a limited dataset in a chosen domain (e.g., medical imaging, remote sensing, gesture recognition, or defect inspection, or any other of your choice).

The main objective is to generate realistic synthetic data that improve downstream model performance in classification or detection tasks, especially when training data are scarce or imbalanced. A secondary objective is to explore domain adaptation between real and synthetic data to improve the generalization capabilities of the classifier (and eventually using a different target domain as test set).

Dataset

May be collected ad hoc or sourced from a public repository containing a small-scale domain-specific dataset. For the dataset, report: Class label and data source; Dataset statistics (counts per class, imbalance ratio, distribution characteristics); Documentation of the domain characteristics and the intended augmentation scope (e.g., intra-class diversity, dataset balancing, etc.). The final dataset must include a reduced baseline subset for initial experiments and an augmented version enriched with synthetic data (and, eventually, a third dataset for domain adaptation).

Experimental Plan

- Phase 1: Train and evaluate a classifier (e.g., CNN or transformer-based model) on the reduced baseline dataset. Record metrics such as accuracy, F1-score, and confusion matrices to quantify the limitations due to data scarcity.
- Phase 2: Implement and train a GAN for generating synthetic samples that mimic the real data distribution. Perform a qualitative and quantitative evaluation of synthetic data.
- Phase 3: Retrain the same classifier on the augmented dataset (real + synthetic data). Compare performance with Phase 1 to assess the contribution of GAN-based augmentation. Optionally, introduce adversarial domain adaptation to improve generalization across domains (i.e., to reduce the effect of domain shift during training & testing).

Point of Contact

Andrea Bottino (andrea.bottino@polito.it).

2.11 Speech Denoising

Objective

Design and implement a deep learning system capable of enhancing speech signals corrupted by an interfering noise. The system must learn to separate clean speech from background or interfering sounds and generalize from synthetic training data to real-world noisy recordings. The project focuses on assessing

how a model trained entirely on simulated mixtures performs on real recordings and exploring strategies to mitigate the performance gap.

Example Datasets

Use publicly available speech datasets to construct the synthetic training corpus, and to evaluate generalization on real noisy speech data. Two distinct datasets must be prepared:

- Synthetic training dataset: obtained by mixing clean speech with noise or interference samples, optionally convolved with Room Impulse Responses to simulate reverberation. The generation process should cover a wide SNR range (e.g., -5 to $+20$ dB) and at least two distinct noise categories.
- Real evaluation dataset: composed of real-world noisy speech recordings that differ in acoustic and recording conditions from the synthetic data.

Each dataset should include: Clean speech signal (e.g., LibriSpeech or LibriVox); Noise signals (e.g., traffic, babble, wind, domestic appliances) from datasets such as FreeSound, MUSAN, ESC-50, UrbanSound8K, DNS Challenge; Metadata (SNR, noise category, and—when applicable—reverberation parameters).

Experimental Plan

- Phase 1: Train a baseline speech enhancement model (e.g., spectral-mapping CNN, LSTM, or U-Net) on the synthetic dataset to predict clean waveform outputs. Evaluate its performance on both synthetic and real noisy speech, quantifying the simulation-to-real gap.
- Phase 2: Introduce and evaluate methods to reduce this gap. Possible strategies include:
 - Data augmentation, diversification and domain randomization, such as broader noise types and microphone/channel simulations.
 - Supervised fine-tuning using a small labeled subset of real noisy/clean pairs.
 - Self-supervised or unsupervised domain adaptation exploiting real noisy recordings without requiring their clean references (e.g., consistency regularization, pseudo-labeling, or adversarial feature alignment). For an overview, see <https://www.v7labs.com/blog/domain-adaptation-guide>.

Example Datasets

LibriSpeech: <http://www.openslr.org/12> LibriVox (not annotated, but with Italian speakers): <https://librivox.org/> UrbanSound8K: <https://urbansounddataset.weebly.com/urbansound8k.html> DEMAND: <https://zenodo.org/records/1227121> MUSAN: <https://www.openslr.org/17/>

Point of Contact

Antonio Seveti (antonio.servetti@polito.it).

2.12 Room Impulse Response Estimation

Objective

Design and implement a machine learning system to estimate the Room Impulse Response (RIR) or a set of its key acoustic parameters (such as Reverberation Time (RT60), Clarity (C50), or room dimensions). The system must take an input signal (e.g., a single-channel reverberant audio recording) and produce the estimated output.

Dataset

The dataset may be sourced from a public repository or simulated, depending on availability and focus. For each instance, provide as input data, the reverberant audio signal, as target output, the full RIR waveform or the ground-truth values for the estimated acoustic parameters (e.g. RT60, room dimensions).

Experimental Plan

- Phase 1: Train and evaluate a baseline neural network model to predict one or more acoustic parameters.
- Phase 2: Explore improvements in model design or target representation such as:
 - Multi-task estimation of several parameters (T60, DFF, C50, EDT).
 - Full or partial RIR reconstruction from time-frequency features.

Example Datasets

BIRD: <https://github.com/FrancoisGrondin/BIRD> BUT ReverbDB: <https://speech.fit.vut.cz/software/but-speech-fit-reverb-database> AIR / OpenAIR / MIRD / MIT IR collections: <https://github.com/RoyJames/room-impulse-responses>

Point of contact

Antonio Seveti (antonio.servetti@polito.it)

2.13 Emotion Recognition from Biometric Signals

Objective

Design and implement a machine learning system capable of recognizing emotional states (positive vs. negative, or specific categories such as amusement,

fear, sadness, etc.) from biometric signals. The goal is to estimate either discrete emotion labels (e.g., “positive emotion”, “negative emotion”) or continuous affective dimensions (e.g., valence and arousal) using physiological data only.

Dataset

Use “Psychophysiology of Positive and Negative Emotions: Dataset of 1157 Cases and 8 Biosignals”, Scientific Data (<https://doi.org/10.1038/s41597-021-01117-0>). It includes 1,157 emotion-eliciting trials from seven studies with healthy participants exposed to stimuli designed to evoke various feelings. For each trial, multiple physiological signals were recorded, such as ECG, ICG, EDA, blood pressure, PPG, respiration, and skin temperature, along with emotional labels indicating the elicited state. The dataset, available at the Psychosensing Data Repository (<https://data.psychosensing.psnc.pl/>), provides a solid foundation for exploring how biometric patterns reflect positive and negative emotions.

Experimental Plan

- Phase 1: Implement and evaluate a baseline unimodal model using a single physiological signal (for example, ECG) to classify positive versus negative emotions. A simple architecture such as an MLP, CNN, or LSTM can be used depending on the signal’s temporal structure.
- Phase 2: Extend the system to multimodal fusion by combining multiple physiological channels (for instance ECG, EDA, and respiration) to capture richer and more complementary emotional patterns. Explore early fusion (feature-level concatenation) and late fusion (decision-level ensemble). Alternatively, integrate temporal modeling techniques such as BiLSTM, or GRU architectures to better capture the dynamic evolution of emotions across time.
- Phase 3: Perform interpretability analysis to identify which physiological features or signal types are most influential in predicting emotions. Methods such as SHAP values, feature importance rankings, or attention-weight visualization can help highlight the contribution of each biosignal.

Point of contact

Alessandro Visconti (visconti.alessandro@polito.it)

Example Datasets

Psychophysiology of Positive and Negative Emotions: Dataset of 1157 Cases and 8 Biosignals, DOI: 10.1038/s41597-021-01117-0. Data repository: <https://data.psychosensing.psnc.pl/>

2.14 Sketch-Based 3D Model Retrieval from a Library

Objective

Design and implement a cross-modal retrieval system that, given a 2D user sketch (e.g., a single-line drawing), retrieves the most visually similar 3D models from a large, predefined library. The system must learn a joint embedding space where 2D sketches and 3D models of the same object category are close to each other.

Dataset

Must use a public benchmark that provides pairs of 2D sketches and corresponding 3D models. The dataset must include: a library of 3D models (e.g., as meshes, point clouds, or multi-view renderings) organized by category; a collection of 2D sketches (as vector or raster images) also mapped to the same categories.

Experimental Plan

- Phase1: Train and evaluate a baseline cross-modal retrieval model using a Siamese architecture with two branches: (i) a CNN for 2D sketches and (ii) a CNN-based encoder for 3D models, implemented either as a multi-view CNN or, in a simplified variant, as a single-view CNN using a canonical rendering per model. Optimize with contrastive or triplet loss (with semi-hard mining) and evaluate on a balanced test set with standard retrieval metrics.
- Phase 2: Propose and evaluate methods to handle the abstraction gap (domain shift). The challenge is that sketches are highly abstract, sparse, and vary in style, while 3D models are realistic and dense. The training data can be manipulated to simulate this gap (e.g., training on “clean” sketches, testing on “messy” user drawings). Propose and evaluate techniques to improve retrieval robustness against sketch ambiguity and abstraction.

Example dataset

SHREC (e.g., SHREC’13, SHREC’14 benchmarks), Sketchfab-based datasets, ShapeNet (paired with sketch datasets like Quick, Draw).

Point of contact

Alberto Cannavò (alberto.cannavo@polito.it).

2.15 Prediction of Cybersickness in Immersive Virtual Reality

Objective

Design and implement a machine learning-based system capable of predicting cybersickness levels experienced by users exposed to immersive virtual environments. The goal is to estimate the onset and intensity of cybersickness from multimodal features derived from user physiological signals (e.g., heart rate, electrodermal activity, EEG) and/or behavioral and visual characteristics of the virtual scene. The system should provide either a continuous prediction (e.g., sickness score) or a discrete label (e.g., none / mild / severe).

Dataset

Use a public dataset such as the one presented in “Cybersickness Prediction Using Deep Neural Networks on Physiological Signals”, or other publicly available VR datasets that include synchronized physiological recordings, motion/interaction data, and subjective sickness scores (e.g., SSQ or similar). For each instance, report the feature modalities used, data source, and sickness labels. Provide overall dataset statistics, including number of subjects, sessions, and sickness-level distribution.

Experimental Plan

- Stage 1: Implement and evaluate a baseline model (e.g., MLP, CNN, or LSTM depending on input modality) trained on physiological or behavioral data alone.
- Stage 2: Introduce multimodal fusion (e.g., combining physiological and visual features) or temporal modeling to capture sickness evolution over time.
- Stage 3: Analyze cross-subject generalization and robustness across sessions or environments. Optionally, perform interpretability analysis (e.g., feature importance, gradient attribution) to identify the most influential predictors of cybersickness.

Evaluation

Provide both quantitative and qualitative analyses: Regression/classification performance metrics (e.g., MAE, RMSE, F1, correlation with SSQ scores). Confusion/error analysis on mispredicted sickness levels. Discussion of generalization limits and possible improvements (e.g., subject adaptation, normalization).

Example Datasets

Savelab: <https://sites.google.com/view/savelab/data-download-links>.

Point of Contact

Davide Calandra (calandra.davide@polito.it)

2.16 Emotion Recognition from Non-Verbal Behavior

Objective

Design and implement a machine learning-based system capable of recognizing human emotions from multimodal non-verbal behavior. The goal is to classify or estimate the emotional state of a virtual human based on his or her behaviors, capturing both semantic and affective cues. The system should be able to output either a discrete emotion label (e.g., happiness, sadness, anger, surprise) or a continuous affective score (e.g., valence/arousal).

Dataset

Use the BEAT: A Large-Scale Semantic and Emotional Multi-modal Dataset for Conversational Gestures Synthesis (https://link.springer.com/chapter/10.1007/978-3-031-20071-7_36). The dataset includes large-scale recordings of conversational behavior with text transcripts, gestures, facial expressions, and emotional annotations. For each instance, report the modalities used (e.g., motion capture gestures), the data sources and the corresponding emotional labels or valence/arousal annotations. Provide dataset statistics, including number of subjects, total duration, modality types, and emotion label distribution.

Experimental Plan

- Stage 1: Implement and evaluate a baseline unimodal model using facial features derived from blendshape coefficients for emotion recognition. The model can employ a simple MLP, CNN, or LSTM architecture trained on facial tracking data to classify basic emotions or predict continuous affective values.
- Stage 2: Introduce multimodal fusion, combining gesture motion data to improve emotion recognition accuracy. Experiment with early fusion (feature concatenation) and late fusion (decision-level aggregation). Alternatively, incorporate temporal modeling (e.g., BiLSTM, Transformer-based architectures) to capture emotion dynamics within conversation turns.
- Stage 3: Conduct interpretability analysis to identify which modalities or features (e.g., prosody, hand movement, or facial expression) contribute most to emotion recognition. Use techniques such as attention visualization or SHAP feature importance.

Evaluation

Provide both quantitative and qualitative analyses.

Point of Contact

Alessandro Visconti (visconti.alessandro@polito.it)