DLSS 2026 – Assignment 2

Social Relation Classification from Images with CNNs

Giordano De Marzo — 23.05.2026

Submission deadline: 17.06.2026 at 23:59

Overview

In this assignment you will use Convolutional Neural Networks (CNNs) to classify the **social relationship** between pairs of people from images. Given a photograph of two individuals, your model must predict whether their relationship is *friends*, *family*, or *professional*.

The dataset is a curated subset of the **People in Social Context (PISC)** dataset (Li et al., ICCV 2017). This is a harder task than standard image benchmarks, so do not expect high accuracy. **A macro-averaged F1 score of 0.50 or above is considered a good result.**

The maximum number of points is **30**. Bonus tasks may award additional points, but the final score is capped at 30.

Dataset

The dataset is available for download from the course page on my website. It contains the following files:

File	Description
`pisc_64x64.zip`	All images under `images/`, JPEG format, 64×64 pixels
`train.csv`	Training split labels
`val.csv`	Validation split labels
`test.csv`	Test split labels

Each CSV has three columns: `filename`, `label_id`, and `label_name`. The three target classes are:

Label ID	Class	Description
0	`friends`	Informal peer relationship
1	`family`	Family members
2	`professional`	Work or institutional relationship

The dataset contains **6,372 training images**, 135 validation images, and 327 test images. The training set is mildly imbalanced: the `professional` class accounts for approximately 43% of training examples.

Important – use the provided splits: Train, validation, and test splits are pre-defined by the CSV files. Do *not* create your own random split. This ensures that your results are comparable across submissions.

Important – image preprocessing: Each image is a 64×64 RGB JPEG. Pixel values should be normalised before training (e.g. to the $[0, 1]$ range or standardised using the dataset mean and standard deviation per channel).

Tasks

The assignment is worth **30 points** divided equally between the code and the written report.

Component	Points
Code	15
Task 1: Data import, visualisation and preprocessing	5
Task 2: Model building and training	5
Task 3: Model evaluation and comparison	5
Report	15

Task 1 Data Import, Visualisation and Preprocessing (5 code points)

- Load the images and labels from the provided CSV files. Verify that filenames and labels align correctly.
- Visualise a random sample of images for each class. Make sure the images and their labels look as expected.
- Analyse and report the class distribution across the train, validation, and test splits. Discuss how the mild imbalance in the training set could affect model training and evaluation, and how you plan to address it (e.g. a class-weighted loss function).
- Preprocess the images: normalise pixel values and apply any other transformations you consider appropriate.
- Describe and justify any data augmentation strategy you use during training (e.g. random horizontal flip, rotation, colour jitter).

Task 2 Model Building and Training (5 code points)

- Implement a **random baseline** that predicts classes uniformly at random (or proportionally to training class frequencies). Report its performance on the test set as a lower bound.
- Implement a **MLP baseline** that operates on flattened images (i.e. no convolutional layers). This illustrates the benefit of spatial feature extraction.
- Build and train **at least two CNN architectures**. Vary at least two design choices across your models, for example the number of convolutional layers, filter sizes, pooling strategy, use of batch normalisation, dropout rate, or learning rate schedule.
- Use a loss function appropriate for multi-class classification. Consider using a class-weighted loss to address the training imbalance.
- Apply appropriate techniques to reduce overfitting.
- Track and plot the learning curves (training loss and validation loss) for each model.

Task 3 Model Evaluation and Comparison (5 code points)

- Evaluate all models (random baseline, MLP, and all CNN variants) on the **validation set**.
- Report meaningful metrics for each model.
- Select your best model and justify the choice. A macro-averaged F1 score of **0.50 or above** is considered a good result for this task.
- Perform the final evaluation of the test set.

Report (15 points)

Write a short scientific report of **at most 3 pages** (excluding bonus tasks). The report should be self-contained: a reader who has not seen this assignment sheet should be able to understand what you did and why. A possible structure is the following:

- **Title, Name and Matriculation Number**
- **Introduction:** briefly introduce the task, the dataset.
- **Results:**
  - *Data:* describe the dataset, the preprocessing and augmentation decisions you made, and any relevant findings from your exploratory analysis (e.g. class distribution, typical visual patterns per class).
  - *Models:* describe the architecture and training setup of each model.
  - *Training:* show and discuss the learning curves for your models.
  - *Evaluation:* Include the confusion matrix and classification report. Show and discuss representative examples of misclassified images.
- **Conclusions:** summarise your findings. Reflect on what visual cues the model may be relying on, and discuss any limitations or directions for future work.

Bonus Tasks

Bonus tasks are optional. You may use up to an additional half page in your report for each bonus task you complete. The final grade is **capped at 30 points**.

Bonus Task 1: Transfer Learning with a Pretrained Model (1 bonus point)

- Fine-tune a pretrained CNN (e.g. ResNet-18, MobileNetV2, or EfficientNet-B0) on the PISC training set. Replace the final classification head with one appropriate for 3-class output and fine-tune the full network or only the last few layers.
- Compare the fine-tuned model with your best from-scratch CNN on the test set. Discuss the trade-offs in terms of performance, training time, and data efficiency.

Bonus Task 2: Visual Interpretability with Gradient Heatmaps (2 bonus points)

- Apply a gradient-based saliency method (e.g. Grad-CAM or Guided Backpropagation) to your best CNN to produce heatmaps highlighting which image regions most influenced predictions.
- Show at least two heatmap examples per class, including at least one correctly classified and one misclassified image.
- Discuss what the heatmaps reveal about the features the model uses to distinguish social relationships. Do the highlighted regions correspond to interpretable visual cues (faces, body language, spatial proximity, background context)?

Submission Guidelines

- Submit your work on the course GitHub as a Colab notebook (`.ipynb`) and a PDF report.
- Your notebook must be clean, well-commented, and fully reproducible. Re-running it from top to bottom should reproduce all results.
- Use a fixed random seed wherever randomness is involved to ensure reproducibility.