# Assignment: Typicality in Deep Neural Representations

## Submission Instructions

Please answer all written questions in a separate document (PDF), formatted using LaTeX or Microsoft Word. Provide short and clear explanations. Your answers should reflect both conceptual reasoning and (where appropriate) references to observed outputs or code behavior.

When writing code, add your code in new code cells inserted into the original notebook at appropriate points. Make sure each code cell is clearly labeled with a comment indicating the question it addresses (e.g., # Part 2, Question 3: Cosine similarity matrix).

Submit the work in the form of a completed Jupyter notebook: (groupXX_typicality_assignment.ipynb) and a PDF with your written answers: (groupXX_typicality_answers.pdf)

## Part 1: Code Understanding

Explain the following with short text. Make sure you can answer these before running the code.

1. **Penultimate layer (fc1):**
   What is the "embedding" extracted from the CNN?
   How is it related to classification?

2. **Logits vs. Softmax:**
   What are some reasons for analyzing both the logits and softmax outputs as measures of typicality? Relate to the work of Lake et al.

3. **Euclidean distance and cosine similarity:**
   What do these metrics capture when evaluating image-embeddings against the average embedding of each category? What is the main difference between them? Which one would you choose when designing a real-world application which recommends to clients 'typical' coats based on a large database of coats? Consider that cosine similarity ignores embedding magnitude. Would you want the coat recommendation-system to prefer larger or more intense feature values, or just directional similarity?

## Part 2: Analysis & Comparison

Run the code and interpret the following outputs:

1. **Compare rankings:**
   Look at the "Spearman Rank Correlation Between Prototypicality Measures". Which measure correlate strongly (e.g., > 0.7) and which weakly? What might be the reason?

2. **Visualization of typical vs. atypical examples:**
   For each typicality method (logit, softmax, distance, cosine), examine the top-5 and bottom-5 images per category. What visual patterns do you notice in the most typical images? What kinds of distortions, ambiguity, or edge cases appear in the least typical images?
3. **Interpret the cosine similarity matrix:**
   In the table, "Cosine similarity between class mean embeddings and classifier weight vectors", you compute the cosine similarity between each class's average embedding and its weight vector in the classifier. What does a high similarity tell you? What would a low similarity mean? Can you think of reasons that produce low similarity?
4. **Study the weight matrix:**
   Implement an analysis that uses only the classifier's weight vectors (i.e., the fc2 weight matrix of shape 10×128) to visualize how the network internally organizes its categories. For example, compute pairwise cosine similarities between the weight vectors and display the result as a heatmap.

**Part 3: Extension.**

Define a new notion of typicality that, for the trained network, considers both:

- Closeness to the image's own class mean (intra-class coherence),
- **And** (choose) one of the following:
    - Distance to the nearest other class mean, or
    - Average distance to all other class means.

The new score should be higher when an image is both similar to its own class mean and far from other class means.

For example, typicality = -alpha * d_self + beta * d_other.

Where *d_self* is distance to own mean, *d_other* is either the average or minimum distance to other class means, and alpha, beta are weights you can set.

Implement this computation for all examples in one category (e.g., "dog"). Compute rank correlation (e.g., Spearman; scipy.stats.spearmanr(score1, score2)) between this new typicality score and the softmax/logit-based scores already computed and evaluate if it produces stronger matches than the other measures already defined in the code.