RESEARCH

Project-9

Sujan Darai (sujad96@zedat.fu-berlin.de), Matanat Mammadli (matanam94@zedat.fu-berlin.de), Samra Hamidovic (samrah96@zedat.fu-berlin.de)

Full list of author information is available at the end of the article

Abstract

Goal of the project: Our project aims to use geometric deep learning for classification, predicting pose or subject ID based on shape differences. We'll employ a transductive learning approach, treating classification as a semi-supervised node labeling problem on a graph. By optimizing a Graph Convolutional Network (GCN), we aim to achieve accurate classification using geometric data.

Main results of the project: This project highlights deeper or stronger connections between the same subjects or poses with moderate clustering coefficients value in the subject-pose graph. However, the KNN graph emphasized the spectral proximity of shapes, forming more isolated clusters and having a higher tendency to cluster with the same poses or subjects than the subject-pose graph.

Personal key learning:

1. Sujan Darai: Got insight about the transductive learning approach
2. Samra Hamidovic: Learned how to create manifold-valued graphs and write reports more professionally.
3. Matanat Mammadli: Learned more about geometric deep learning and shape analysis, also how to plot manifold-valued graphs.

Estimated working hours:

1. Sujan Darai: 8 hours
2. Samra Hamidovic: 8 hours
3. Matanat Mammadli: 8 hours

Project evaluation: 1

Number of words: 1566

Keywords: Graph Convolution Network (GCN), Transductive learning, Manifold-values graphs, ShapeDNA

1 Introduction

Shape analysis and geometric deep learning are essential for understanding complex geometric data, particularly in fields like medical imaging and disease classification. Manifold-valued graphs provide a framework for representing shapes on non-linear manifolds, with nodes encapsulating geometric properties and relationships. Techniques such as k-Nearest-Neighbor graphs based on ShapeDNA descriptors enhance the spectral representation of shape variances, ensuring a robust analysis of intricate geometric data while preserving their inherent structure and relationships. [2] To address classification challenges, transductive learning methods are explored, treating them as semi-supervised node labeling problems on graphs where shape differences serve as key features. This involves examining graph properties and structural characteristics to gain insights into the connectivity and significance of

shapes within datasets, thereby improving the understanding of shape interconnections and their relevance to classification. [2]
Graph Convolutional Networks (GCNs) have become powerful tools for analyzing graph-structured data. Utilizing GCNs in a transductive design shows significant potential across various applications. By incorporating modules such as FlowLayer, TangentMLP, and MfdInvariant from morphomatics.nn, GCNs can effectively process information flow, perform non-linear transformations, and enforce manifold invariance properties. The goal is to classify subjects based on shape differences using population graphs. Training the GCN model on these graphs and evaluating its classification performance across different populations aims to assess the model's generalization capabilities and its ability to capture shape variances specific to each population. This integration of geometric deep learning and GCNs aims to advance image analysis and enhance classification accuracy and robustness in complex geometric datasets. [1]

2 Goal of the project

Our project aims to utilize the potential of geometric deep learning techniques for classification tasks. Specifically, we will employ models that capitalize on shape differences identified in earlier stages to predict attributes such as pose or subject ID. The approach centers around implementing a transductive learning method, treating classification as a semi-supervised node labeling problem on a graph. Here, the nodes represent instances characterized by their shape differences, which serve as intrinsic features. A key component of our methodology involves the deployment and optimization of a Graph Convolutional Network (GCN). GCNs are well-suited for extracting insights from graph-structured data, making them ideal for our application where shape differences naturally form nodes within the graph. By integrating geometric deep learning principles with GCN capabilities, our objective is to achieve precise and resilient classification outcomes. This project aims to push the boundaries of image analysis by exploring innovative ways to harness geometric data for classification purposes.

3 Data and preprocessings

Like in the last two projects, we worked with the FAUST shape dataset (https:// nuage.lix.polytechnique.fr/index.php/s/LJFXrsTG22wYCXx/download?path=%2F&files=FAUST_r.zip). FAUST comes with high-quality correspondences that serve as ground truth. FAUST contains 300 real human scans of 10 different subjects in 30 different poses, acquired with a high-accuracy 3D multi-stereo system. In this report, we have used only two poses of the Faust dataset i.e. hand stretching upward and downward. The poses with the human shape with hand downward were labeled source while the shape with hand upward was labeled as target.
To recover correspondence for remeshed versions of the original data, we went through the steps according to the paper [3]. The dataset was already cleaned and thus no further preprocessing was required. It was then loaded in the jupyter notebook or google collab and tasks were done.

4 Methods

4.1 Manifold-valued Graphs

In the context of shape analysis and geometric deep learning, manifold-valued graphs play a crucial role in representing shapes or geometric data that lie on a non-linear manifold. Nodes in these graphs represent shapes, while edges capture connections based on shared characteristics such as the same subject or similar poses. The manifold-valued nature of the nodes reflects the intrinsic geometric properties of the shapes being studied. Additionally, utilizing a k-nearest-neighbor graph based on ShapeDNA descriptors allows for capturing spectral distances between shapes, enhancing the representation of shape relationships. We will explore a transductive learning method that formulates the classification task as a semi-supervised node labeling problem on a graph with shape differences as node features. Investigating graph properties such as the number of components, clustering coefficient, and degree centrality provides valuable insights into the structural characteristics of these manifold-valued graphs, shedding light on the connectivity and importance of shapes within the dataset. This approach offers a powerful framework for analyzing complex geometric data while preserving their underlying geometric structure and relationships. [2]

4.2 Graph convolutional network

A graph convolutional network (GCN) is a neural network designed for graph-structured data, operating in a transductive design. Modules like FlowLayer, TangentMLP, and MfdInvariant from morphomatics.nn are utilized to manage information flow, perform non-linear transformations, and enforce invariance properties with respect to the manifold. By training the GCN on population graphs representing shape differences and comparing its classification performance across different populations, the model's generalization capabilities and effectiveness in capturing shape variances specific to each population are evaluated. This application of GCNs in a transductive learning setting showcases their potential for classifying subjects based on shape differences in graph representations. [1]

5 Results and discussion

The objective of our analysis was to construct and investigate two types of graphs based on the FAUST dataset: the Subject-Pose Graph (see Figure 1) and the k-nearest-neighbor (k-NN) Graph (see Figure 2). These graphs were intended to explore the relationships between 3D shapes of human bodies based on their subject and pose, and their spectral properties respectively.

We started by installing the necessary dependencies and downloading the FAUST dataset. Each shape in the dataset was processed to compute its manifold harmonics, which are the eigenvalues and eigenvectors of the Laplacian matrix of the shape's mesh. Using the computed manifold harmonics, we created functional maps between shapes. These maps were refined using the ZoomOut algorithm, which iteratively improved the correspondence between the spectral properties of different shapes.

The graphs were visualized using a spring layout to facilitate the interpretation of their structure and properties.

For the Subject-Pose graph, we constructed a graph where each node represents a shape, identified by a tuple of subject ID and pose. Edges were added between nodes if they belonged to the same subject or had the same pose, capturing the relationships based on these two criteria. The plot of the Subject-Pose Graph (see Figure 1) displayed clear clusters of nodes, highlighting the dense connections among shapes of the same subject or pose. These clusters visually affirmed the high clustering coefficient and the well-connected nature of the graph.

For the k-nearest-neighbor (k-NN) graph, another graph was constructed where nodes again represented shapes. Here, edges connected each shape to its k-nearest neighbors based on the distance between their spectral properties (ShapeDNA). The k-NN Graph plot (see Figure 2) showed a more dispersed structure with connections highlighting the spectral similarity between shapes. Unlike the Subject-Pose Graph, this graph emphasized the spectral proximity of shapes, forming more isolated clusters.

For both graphs, we analyzed several properties: Number of Components (Indicates the number of disconnected subgraphs), Clustering Coefficient (Measures the degree to which nodes tend to cluster together), Degree Centrality (Represents the centrality of nodes based on their degree, or number of connections). Both results can be observed in the tables described below (see tablke 1 and table 2).

The Subject-Pose Graph (see table 1) had only 1 component, indicating that all nodes (shapes) were interconnected either directly or indirectly. This graph exhibited a clustering coefficient of approximately 0.471, indicating a moderate degree of clustering among nodes. This reflects the inherent similarity between shapes of the same subject or pose. The degree centrality for all nodes was uniformly 0.202, indicating that each node had the same number of connections. This is because each shape was connected to all other shapes with the same subject or pose, creating a balanced structure.

The k-NN Graph (see table 2) had 3 components, reflecting the sparser connections based only on spectral similarity. The clustering coefficient here was higher at approximately 0.645, indicating that the k-NN Graph had a higher tendency for nodes to cluster than the Subject-Pose Graph. The degree of centrality varied among nodes, with some nodes having higher centrality due to their many spectrally similar neighbors. This variability highlighted the differences in spectral properties among shapes.

For Task 2, we could not produce the results with the provided code.

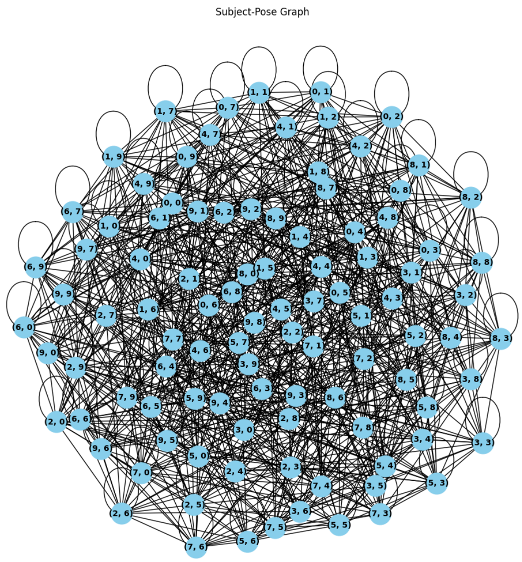

Figure 1: Subject-pose graph of shapes which are of the same subject or are in the same pose

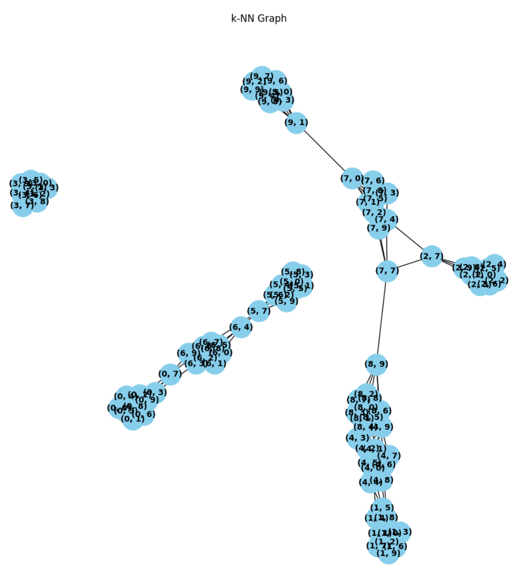

Figure 2: k-Nearest-Neighbor graph using the distance of spectra known as ShapeDNA

Subject-Pose Graph Properties:
{'num_components': 1, 'clustering_coefficient': 0.47058823529411653,
'degree_centrality': {(0, 0)
0.2020202020202020204, (0, 1): 0.2020202020202020204, (0, 2): 0.2020202020202020204, (0, 3):
0.2020202020202020204, (0, 4): 0.2020202020202020204, (0, 5): 0.2020202020202020204, (0, 6):
0.2020202020202020204, (0, 7): 0.2020202020202020204, (0, 8): 0.2020202020202020204, (0, 9):
0.2020202020202020204, (1, 0): 0.2020202020202020204, (2, 0): 0.2020202020202020204, (3, 0):
0.2020202020202020204, (4, 0): 0.2020202020202020204, (5, 0): 0.2020202020202020204, (6, 0):
0.2020202020202020204, (7, 0): 0.2020202020202020204, (8, 0): 0.2020202020202020204, (9, 0):
0.2020202020202020204, (1, 1): 0.2020202020202020204, (2, 1): 0.2020202020202020204, (3, 1):
0.2020202020202020204, (4, 1): 0.2020202020202020204, (5, 1): 0.2020202020202020204, (6, 1):
0.2020202020202020204, (7, 1): 0.2020202020202020204, (8, 1): 0.2020202020202020204, (9, 1):
0.2020202020202020204, (1, 2): 0.2020202020202020204, (2, 2): 0.2020202020202020204, (3, 2):
0.2020202020202020204, (4, 2): 0.2020202020202020204, (5, 2): 0.2020202020202020204, (6, 2):
0.2020202020202020204, (7, 2): 0.2020202020202020204, (8, 2): 0.2020202020202020204, (9, 2):
0.2020202020202020204, (1, 3): 0.2020202020202020204, (2, 3): 0.2020202020202020204, (3, 3):
0.2020202020202020204, (4, 3): 0.2020202020202020204, (5, 3): 0.2020202020202020204, (6, 3):
0.2020202020202020204, (7, 3): 0.2020202020202020204, (8, 3): 0.2020202020202020204, (9, 3):
0.2020202020202020204, (1, 4): 0.2020202020202020204, (2, 4): 0.2020202020202020204, (3, 4):
0.2020202020202020204, (4, 4): 0.2020202020202020204, (5, 4): 0.2020202020202020204, (6, 4):
0.2020202020202020204, (7, 4): 0.2020202020202020204, (8, 4): 0.2020202020202020204, (9, 4):
0.2020202020202020204, (1, 5): 0.2020202020202020204, (2, 5): 0.2020202020202020204, (3, 5):
0.2020202020202020204, (4, 5): 0.2020202020202020204, (5, 5): 0.2020202020202020204, (6, 5):
0.2020202020202020204, (7, 5): 0.2020202020202020204, (8, 5): 0.2020202020202020204, (9, 5):
0.2020202020202020204, (1, 6): 0.2020202020202020204, (2, 6): 0.2020202020202020204, (3, 6):
0.2020202020202020204, (4, 6): 0.2020202020202020204, (5, 6): 0.2020202020202020204, (6, 6):
0.2020202020202020204, (7, 6): 0.2020202020202020204, (8, 6): 0.2020202020202020204, (9, 6):
0.2020202020202020204, (1, 7): 0.2020202020202020204, (2, 7): 0.2020202020202020204, (3, 7):
0.2020202020202020204, (4, 7): 0.2020202020202020204, (5, 7): 0.2020202020202020204, (6, 7):
0.2020202020202020204, (7, 7): 0.2020202020202020204, (8, 7): 0.2020202020202020204, (9, 7):
0.2020202020202020204, (1, 8): 0.2020202020202020204, (2, 8): 0.2020202020202020204, (3, 8):
0.2020202020202020204, (4, 8): 0.2020202020202020204, (5, 8): 0.2020202020202020204, (6, 8):
0.2020202020202020204, (7, 8): 0.2020202020202020204, (8, 8): 0.2020202020202020204, (9, 8):
0.2020202020202020204, (1, 9): 0.2020202020202020204, (2, 9): 0.2020202020202020204, (3, 9):
0.2020202020202020204, (4, 9): 0.2020202020202020204, (5, 9): 0.2020202020202020204, (6, 9):
0.2020202020202020204, (7, 9): 0.2020202020202020204, (8, 9): 0.2020202020202020204, (9, 9):
0.2020202020202020204}}

k-NN Graph Properties:
{'num_components': 3, 'clustering_coefficient': 0.6454725829725828,
'degree_centrality': {(0, 0):
0.05050505050505051, (0, 5): 0.06060606060606061, (0, 8): 0.08080808080808081, (0, 1):
0.05050505050505051, (0, 4): 0.05050505050505051, (0, 2): 0.08080808080808081, (0, 3):
0.07070707070707072, (0, 6): 0.05050505050505051, (0, 9): 0.05050505050505051, (6, 3):
0.09090909090909091, (0, 7): 0.05050505050505051, (6, 9): 0.05050505050505051, (6, 7):
0.08080808080808081, (1, 0): 0.08080808080808081, (1, 5): 0.07070707070707072, (1, 4):

0.07070707070707072, (1, 1): 0.05050505050505051, (1, 8): 0.11111111111111112, (1, 6): 0.06060606060606061, (1, 2): 0.06060606060606061, (1, 7): 0.05050505050505051, (1, 3): 0.05050505050505051, (4, 0): 0.11111111111111112, (1, 9): 0.05050505050505051, (4, 8): 0.06060606060606061, (2, 0): 0.08080808080808081, (2, 5): 0.06060606060606061, (2, 2): 0.06060606060606061, (2, 8): 0.08080808080808081, (2, 3): 0.06060606060606061, (2, 1): 0.08080808080808081, (2, 6): 0.06060606060606061, (2, 9): 0.06060606060606061, (2, 4): 0.05050505050505051, (2, 7): 0.05050505050505051, (7, 7): 0.05050505050505051, (7, 4): 0.08080808080808081, (3, 0): 0.08080808080808081, (3, 5): 0.08080808080808081, (3, 1): 0.08080808080808081, (3, 3): 0.06060606060606061, (3, 2): 0.07070707070707072, (3, 6): 0.09090909090909091, (3, 9): 0.06060606060606061, (3, 4): 0.06060606060606061, (3, 8): 0.07070707070707072, (3, 7): 0.05050505050505051, (4, 5): 0.08080808080808081, (4, 1): 0.11111111111111112, (4, 2): 0.05050505050505051, (8, 4): 0.10101010101010102, (4, 4): 0.05050505050505051, (8, 5): 0.08080808080808081, (8, 1): 0.07070707070707072, (4, 9): 0.05050505050505051, (4, 3): 0.05050505050505051, (8, 3): 0.07070707070707072, (8, 7): 0.06060606060606061, (4, 6): 0.06060606060606061, (8, 6): 0.06060606060606061, (4, 7): 0.05050505050505051, (8, 9): 0.06060606060606061, (5, 0): 0.08080808080808081, (5, 5): 0.07070707070707072, (5, 8): 0.06060606060606061, (5, 1): 0.07070707070707072, (5, 4): 0.07070707070707072, (5, 6): 0.08080808080808081, (5, 3): 0.05050505050505051, (5, 9): 0.06060606060606061, (5, 2): 0.05050505050505051, (5, 7): 0.06060606060606061, (6, 4): 0.05050505050505051, (6, 0): 0.07070707070707072, (6, 5): 0.08080808080808081, (6, 1): 0.05050505050505051, (6, 8): 0.07070707070707072, (6, 6): 0.07070707070707072, (6, 2): 0.05050505050505051, (7, 0): 0.07070707070707072, (7, 5): 0.08080808080808081, (7, 8): 0.07070707070707072, (7, 1): 0.06060606060606061, (9, 1): 0.06060606060606061, (7, 2): 0.08080808080808081, (7, 9): 0.05050505050505051, (7, 3): 0.05050505050505051, (7, 6): 0.05050505050505051, (8, 0): 0.07070707070707072, (8, 8): 0.09090909090909091, (8, 2): 0.07070707070707072, (9, 0): 0.07070707070707072, (9, 4): 0.09090909090909091, (9, 5): 0.09090909090909091, (9, 3): 0.06060606060606061, (9, 8): 0.08080808080808081, (9, 2): 0.05050505050505051, (9, 7): 0.05050505050505051, (9, 9): 0.05050505050505051, (9, 6): 0.05050505050505051}}

6 Contributions

Sujan Darai: Running and Implementing Task 1, and Task 2 and overviewing and correcting the report in general
Matanat Mammadli: Running and Implementing Task 1, Running Task 2, Writing Results & Discussion
Samra Hamidovic: Running and Implementing Task 1, Running Task 2, Writing Introduction, Goal of the project, Data & preprocessing, Methods

7 Appendix

References

1. Martin Hanik, Gabriele Steidl, and Christoph von Tycowicz. Manifold GCN: Diffusion-based Convolutional Neural Network for Manifold-valued Graphs, January 2024. arXiv:2401.14381 [cs, math].
2. Julius Mayer, Daniel Baum, Felix Ambellan, and Christoph von von Tycowicz. A Soft-Correspondence Approach to Shape-based Disease Grading with Graph Convolutional Networks. In *Proceedings of the First International Workshop on Geometric Deep Learning in Medical Image Analysis*, pages 85–95. PMLR, November 2022. ISSN: 2640-3498.
3. Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics*, 31(4):30:1–30:11, July 2012.