Integrating Large Language Models and Graph Convolutional Networks for Semi-Supervised Image Classification

Camila Piscioneri Magalhães[1], **Lucas Pascotti Valem**[1]

[1] Institute of Mathematics and Computer Science (ICMC)
University of São Paulo (USP)
São Carlos – SP – Brazil

piscioneri@usp.br, lucas@icmc.usp.br

Abstract. *While the growing availability of image data has driven significant advances, labeling datasets remains costly and time-consuming. Therefore, semi-supervised approaches such as Graph Convolutional Networks (GCNs), which learn from both labeled and unlabeled data, have emerged as a promising solution. One of the primary challenges in applying GCNs to image classification is graph construction, since, unlike in citation networks or similar domains, images typically do not come with a predefined structural representation. For visual data, most studies construct graphs based on the similarity between feature vectors from pretrained deep learning backbones, typically by employing kNN or reciprocal kNN algorithms. Although Large Language Models (LLMs) have shown remarkable capability in capturing high-level semantics, their integration with GCNs for image classification remains underexplored. Aiming to fill this gap, our approach uses a Vision Language Model (VLM) to generate textual image descriptions, which are then processed by an LLM to estimate semantic similarity scores between connected images. These scores guide the pruning of edges in kNN and reciprocal kNN graphs, filtering out semantically irrelevant neighbors. Experimental results reveal that leveraging LLMs for graph refinement can improve classification accuracy, particularly for kNN graphs and some backbones. The source code is publicly available at* `gcnllm.lucasvalem.com`.

1. Introduction

The volume of image data being produced and stored has grown dramatically in recent years, driven by applications in areas such as healthcare, security, and social media [Uelwer et al. 2025, Ghosh et al. 2024]. Organizing, indexing, and classifying these large image collections is a central problem for modern data management systems. While collecting images has become inexpensive, labeling them remains costly and time-consuming, as it typically depends on specialized human effort [Uelwer et al. 2025]. This limitation motivates methods that learn effectively from collections in which only a small fraction of the samples is labeled.

In this scenario, semi-supervised methods, which learn jointly from labeled and unlabeled data, have emerged as a promising direction. Among them, Graph Convolutional Networks (GCNs) [Wu et al. 2019] stand out by representing samples as nodes of a graph and propagating label information through the connections between them, allowing the few available labels to guide the classification of the unlabeled ones [Müller et al. 2024].

A key obstacle when applying GCNs to image classification is that, unlike citation networks and other naturally graph-structured domains, image collections usually do not come with a predefined graph: it must be constructed from the data itself. Most approaches build this graph from the similarity between feature vectors extracted by pretrained deep networks, typically using kNN or reciprocal kNN graphs [Müller et al. 2024]. However, how to model an effective graph remains an open research challenge, since proximity in the visual feature space does not always reflect semantic relationships and may introduce noisy connections that degrade classification.

Recently, Large Language Models (LLMs) have shown a remarkable ability to capture high-level semantics from text [Li et al. 2024, Ghosh et al. 2024], yet their use to support graph construction for image classification is still underexplored [Li et al. 2024]. The objective of this work is to investigate this combination: a Vision Language Model (VLM) generates textual descriptions of the images, and an LLM estimates the semantic similarity between connected images, producing scores that prune semantically irrelevant edges from kNN and reciprocal kNN graphs. We present this approach together with an experimental evaluation on the Corel5k dataset, whose encouraging preliminary results indicate that LLM-based graph refinement can improve classification accuracy and point to a promising direction for ongoing research.

2. Proposed Approach

Figure 1 presents an overview of the proposed method for semi-supervised image classification with GCNs, where an LLM refines the graph based on textual descriptions generated from the images by a VLM. The main stages of the process are described in the following subsections.

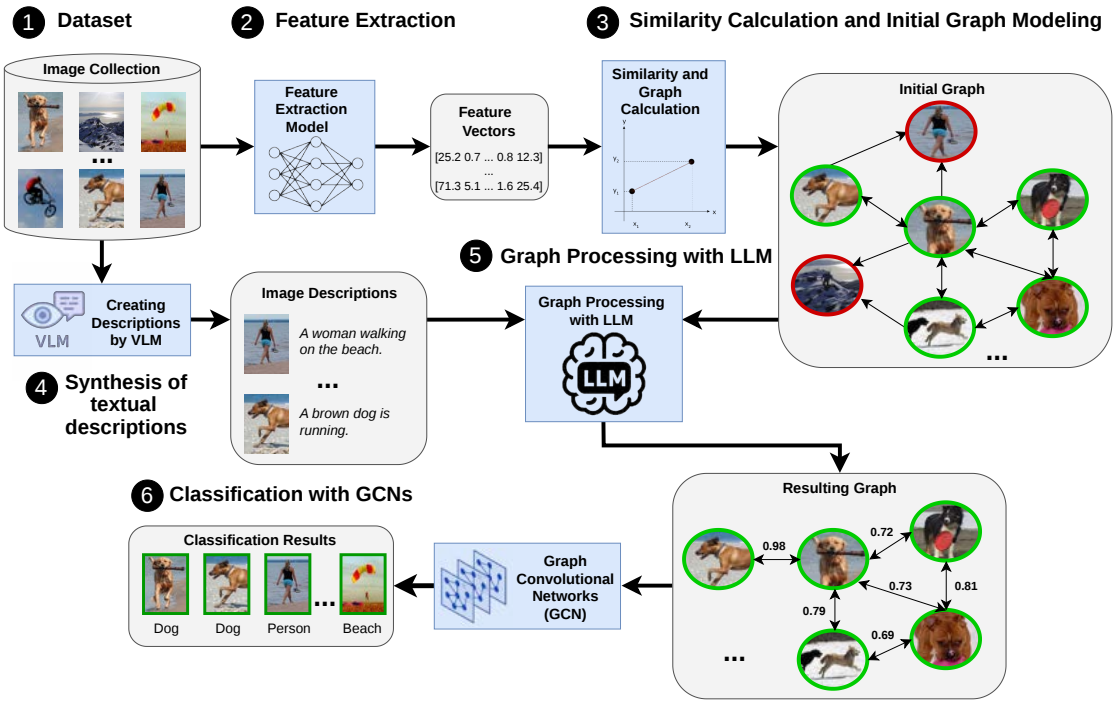


Figure 1. Overview of the proposed approach integrating LLMs for graph refinement with GCNs for semi-supervised image classification.

2.1. Dataset and Feature Extraction

The proposed approach is presented using the Corel5k dataset [Liu and Yang 2013], which consists of 5,000 images distributed across 50 classes. Since this dataset does not provide associated textual descriptions, the captions required by the method are generated using a VLM according to the process described in Section 2.3.

For visual feature extraction, the approach employs three pretrained deep learning models: ResNet [He et al. 2016], the Vision Transformer (ViT) [Dosovitskiy et al. 2021], and DINOv2 [Oquab et al. 2024]. Each model encodes every image into its own feature vector, and the resulting vectors are used to build a distinct input graph per extractor for the semi-supervised classification with the GCN.

2.2. Graph Modeling

Let x_i and x_j be two images that may or may not be connected. From the feature vectors extracted as described in the previous section, we use a Ball Tree to efficiently retrieve, based on the Euclidean distance, the set of the top-k nearest neighbors of x_i, denoted by $\mathcal{N}(x_i, k)$. We consider two common graph construction strategies, which serve both as baselines and as a basis for investigating the use of LLMs: the kNN graph and the reciprocal kNN graph [Valem et al. 2023, Yang et al. 2025].

In the kNN graph, each image is connected to its k nearest neighbors, producing the edge set $E_{knn} = \{(i, j) \mid j \in \mathcal{N}(x_i, k)\}$. In the reciprocal kNN graph, an edge is created only when the neighborhood relationship is mutual, that is, $E_{rec} = \{(i, j) \mid j \in \mathcal{N}(x_i, k) \wedge i \in \mathcal{N}(x_j, k)\}$. The resulting graph can then be refined by the LLM before being provided as input to the GCN.

2.3. Synthesis of Textual Descriptions

Since the Corel5k dataset does not provide textual descriptions associated with the images, a VLM was adopted to generate the captions. In this work, the BLIP (*Bootstrapping Language-Image Pre-training*) model [Li et al. 2022] was employed due to its ability to produce textual descriptions from image content. The process consists of providing an image as input to the model, which generates a corresponding caption, as illustrated in step 4 of Figure 1. These captions are then employed as textual representations to formulate the input for the LLM during the graph refinement step.

2.4. Graph Processing with LLM

After computing the initial graph as described in Section 2.2, a semantic refinement step is performed using an LLM. For this stage, we adopt the GPT-OSS-20B model, accessed through the Groq platform[1]. This model was chosen because it is a publicly available, open-weight LLM of moderate size, which makes our approach reproducible and computationally accessible without relying on large-scale proprietary models. For reproducibility, all queries used a random seed of 1000, temperature of 0, and a maximum generation length of $10{,}000$ tokens.

In this stage, the textual descriptions associated with the images are used to estimate the semantic similarity between connected image pairs, as illustrated in step 5 of Figure 1. Given the top-k elements of each image provided by a feature extractor (i.e., DINOv2, ResNet, and ViT), we formulate a prompt to compute the semantic similarity

[1]Online API that allows the execution of open-source LLMs: `https://groq.com/`.

between images, as presented in Listing 1. This prompt instructs the LLM to score how semantically similar each candidate caption is to a reference caption, returning only the numeric scores in the range $[0, 1]$. The reference caption corresponds to the query image, while the candidate captions are the captions of its neighbors, sorted in descending order of similarity. This process is repeated for every image to obtain the full set of scores.

The similarity scores generated by the LLM are used to filter connections in the initial graph, retaining only edges whose semantic similarity exceeds a threshold th, resulting in the following: $E_{LLM} = \{(i, j) \mid (i, j) \in E_{initial} \wedge \rho(i, j) \geq th\}$, where $\rho(i, j)$ represents the similarity estimated by the LLM between images i and j. Note that $E_{initial}$ can be either E_{knn} or E_{rec}.

Listing 1. Prompt used for semantic similarity evaluation.

```
You are evaluating visual semantic similarity.
Reference: {reference_caption}
Step 1: Identify:
* main subject
* key attributes (clothing, objects, actions)
Step 2: Score each phrase:
Scoring rules:
1.0     = same subject + same key attributes
0.8-0.9 = same subject + at least one key attribute
0.5-0.7 = same subject only
0.1-0.4 = weak relation (very generic)
0.0     = different subject or unrelated
Important:
* Focus on visual meaning
* Be consistent across all phrases
Phrases:
1. {neighbor_caption_1}
2. {neighbor_caption_2}
...
k. {neighbor_caption_k}
Output ONLY the scores separated by spaces.
```

2.5. Semi-Supervised Classification with Graph Convolutional Networks

The classification stage was performed using the *Simple Graph Convolution* (SGC) model [Wu et al. 2019]. By removing nonlinearities and collapsing the weight matrices into a single linear transformation, SGC is a lightweight and efficient model. This makes it well suited to validate the proposed approach with minimal architectural overhead. The model receives as input the image feature vectors and the graph structure constructed in the previous stage, enabling the integration of visual information and semantic relationships between nodes during the classification process. The idea is that pruning low-similarity edges with an LLM filters out unreliable connections produced by purely visual features, providing a graph that improves classification effectiveness.

3. Experimental Results

Two graph construction strategies were evaluated, one based on kNN and another on reciprocal kNN, both with $k = 20$ and including their variants with LLM-based semantic refinement, which is our proposed approach. All configurations were assessed using the SGC model for classification under a reverse stratified 10-fold cross-validation protocol in which only 10% is used for training and 90% for testing to assess effectiveness under

limited supervision, with classification accuracy adopted as the evaluation measure. All results report the mean and standard deviation of 10 runs.

To analyze the sensitivity of the refinement step, Figure 2 presents, for each feature and graph combination, a dashed line that indicates the baseline (no LLM) and a curve that shows how the LLM refinement impacts the GCN accuracy as the threshold varies. As can be seen, 0.2 is a good default value in most cases, and it is therefore adopted in the remaining experiments.

Building on this choice, Table 1 presents the results obtained for different graphs, reporting both the default threshold of 0.2 and the best threshold evaluated. Overall, semantic edge refinement based on LLMs improved accuracy in most cases when compared to graphs built solely from visual similarity. The improvements are concentrated on the kNN graphs, whose purely visual neighborhoods are often noisier and benefit most from the LLM. In contrast, the reciprocal kNN graphs already provide stronger baselines, so the LLM provides smaller gains. A clear saturation case is observed for ViT, which is already a strong feature extractor.

These results suggest that LLM refinement is most effective on noisier graph structures, while providing smaller contributions when the underlying graph is already highly discriminative. Although this behavior deserves further investigation, the consistent gains observed across the noisier configurations highlight the strong potential of LLMs as a flexible tool for enhancing graph-based learning in computer vision tasks, particularly using GCN models. We believe that investigating different prompts, parameters, and LLM models can lead to more improvements, opening promising directions for future research.

Table 1. Classification accuracy (%) with and without LLM graph refinement.

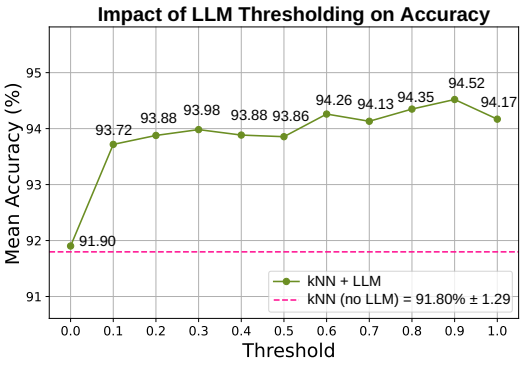
Graph	Method	Feature Extractor		
		DINOv2	**ViT**	**ResNet**
kNN	Baseline (no LLM)	91.80 ± 1.29	92.80 ± 0.38	89.04 ± 0.75
	+ LLM (default threshold=0.2)	93.88 ± 1.26	94.12 ± 0.48	91.22 ± 0.60
	+ LLM (with best threshold)	94.52 ± 1.06 (th=0.9)	94.12 ± 0.48 (th=0.2)	91.23 ± 0.74 (th=0.5)
	Relative gain (%)	**+2.96%**	**+1.42%**	**+2.46%**
Rec. kNN	Baseline (no LLM)	94.90 ± 0.78	95.33 ± 0.45	92.04 ± 0.79
	+ LLM (default threshold=0.2)	95.03 ± 0.85	94.98 ± 0.39	92.56 ± 0.55
	+ LLM (with best threshold)	95.08 ± 0.88 (th=0.4)	95.32 ± 0.46 (th=0.0)	92.57 ± 0.51 (th=0.1)
	Relative gain (%)	**+0.19%**	**−0.01%**	**+0.58%**

4. Conclusion and Future Work

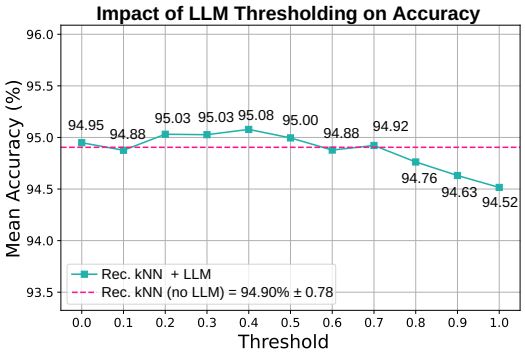
This work investigated the use of LLMs to refine graphs for semi-supervised image classification with GCNs. Textual descriptions generated by a VLM are processed by an LLM to estimate the semantic similarity between connected images, pruning edges that are inconsistent in graphs built solely from visual features. Experiments indicate that this refinement can indeed improve accuracy. However, graphs that are already very effective show smaller or no benefit from this strategy, requiring further investigation. As a continuation of this research, we intend to explore different *prompt engineering* strategies, additional LLMs, parameter configurations, and other image datasets to assess our approach, as well as more adaptive ways of combining visual and semantic similarity beyond a fixed threshold.

Origin of the Work

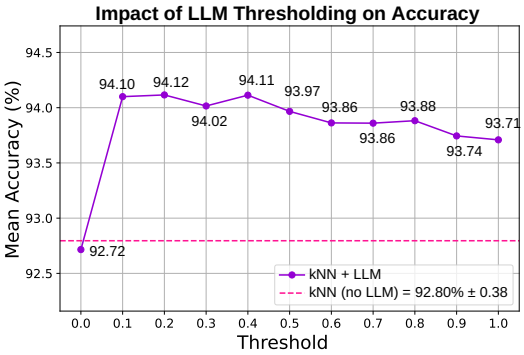
This paper presents ongoing work from an undergraduate research project conducted by the student Camila Piscioneri Magalhães.



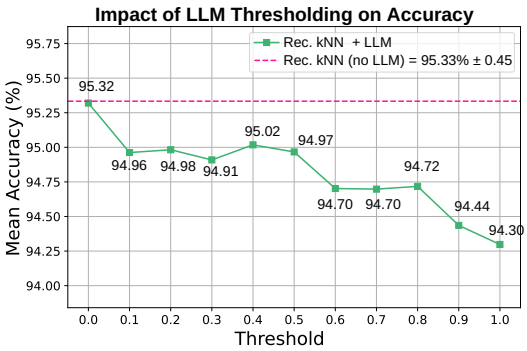
(a) DINOv2 with kNN graph



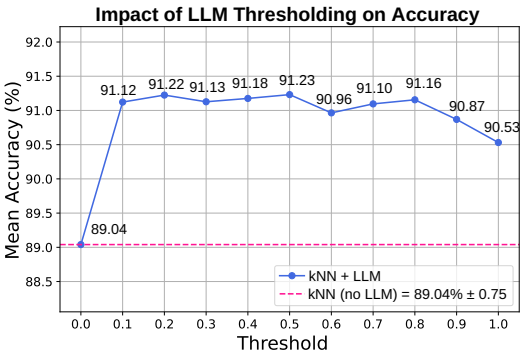
(b) DINOv2 with Rec. kNN graph



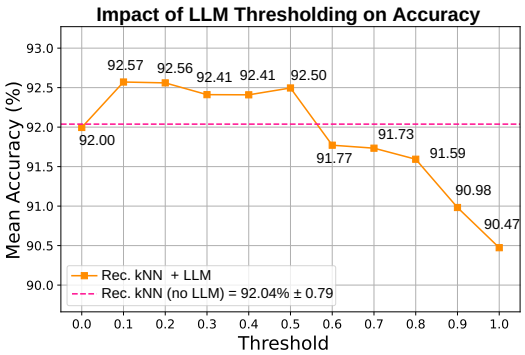
(c) ViT with kNN graph



(d) ViT with Rec. kNN graph



(e) ResNet with kNN graph



(f) ResNet with Rec. kNN graph

Figure 2. Impact of LLM thresholding on SGC classification accuracy.

Acknowledgments

Camila Piscioneri Magalhães is supported by a fellowship from the *Programa Unificado de Bolsas* (PUB/USP). This work was also financially supported by the São Paulo Research Foundation (FAPESP, grant #2025/10602-5), the University of São Paulo (USP, PRPI Ordinance No. 1032, "*Apoio aos Novos Docentes*"), and the Institute of Mathematics and Computer Science (ICMC-USP).

AI Usage Declaration

The authors used AI-based language models (Claude and Gemini) exclusively for reviewing and improving the clarity of text written by the authors.

References

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

Ghosh, A., Acharya, A., Saha, S., Jain, V., and Chadha, A. (2024). Exploring the frontier of vision-language models: A survey of current methodologies and future directions. *arXiv preprint arXiv:2404.07214*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Li, J., Li, D., Xiong, C., and Hoi, S. C. H. (2022). Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*.

Li, Y., Li, Z., Wang, P., Li, J., Sun, X., Cheng, H., and Yu, J. X. (2024). A survey of graph meets large language model: Progress and future directions. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*.

Liu, G.-H. and Yang, J.-Y. (2013). Content-based image retrieval using color difference histogram. *Pattern Recognition*, 46(1):188 – 198.

Müller, T. T., Starck, S., Dima, A., Wunderlich, S., Bintsi, K.-M., Zaripova, K., Braren, R. F., Rückert, D., Kazi, A., and Kaissis, G. (2024). A survey on graph construction for geometric deep learning in medicine: Methods and recommendations. *Transactions on Machine Learning Research*.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jégou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. (2024). Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*.

Uelwer, T., Robine, J., Wagner, S. S., Höftmann, M., Upschulte, E., Konietzny, S., Behrendt, M., and Harmeling, S. (2025). A survey on self-supervised methods for visual representation learning. *Machine Learning*, 114(4):111.

Valem, L. P., Pedronette, D. C. G., and Latecki, L. J. (2023). Graph convolutional networks based on manifold learning for semi-supervised image classification. *Computer Vision and Image Understanding*, 227:103618.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. (2019). Simplifying graph convolutional networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6861–6871. PMLR.

Yang, J., Li, H., Du, B., and Ye, M. (2025). Cheb-gr: Rethinking k-nearest neighbor search in re-ranking for person re-identification. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19261–19270.