Image classification by Distortion-Free Graph Embedding and KNN-Random forest

1st Askhat Temir
Department of Computer Science
Nazarbayev University
Nur-Sultan, Kazakhstan
askhat.temir@nu.edu.kz

2nd Kamalkhan Artykbayev
Department of Computer Science
Nazarbayev University
Nur-Sultan, Kazakhstan
kamalkhan.artykbayev@nu.edu.kz

3rd M. Fatih Demirci
Department of Computer Science
Nazarbayev University
Nur-Sultan, Kazakhstan
muhammed.demirci@nu.edu.kz

***Abstract*—Image classification algorithms play an important role in various computer vision problems such as object tracking, image labeling, and object segmentation. A number of methodologies have been proposed to tackle this problem. One of the possible approaches employed extensively in the literature is to represent an image as a graph based on its hand-crafted features. However, recent advancements in deep neural networks have shown their ability to learn more discriminative and representative features. Therefore, the deep features have become considerable alternatives of hand-crafted ones. In this paper, we propose a novel framework based on distortion-free graph embedding using deep features and KNN-Random forest. Our method outperforms the state-of-the-art graph embedding-based image classification approach for the task of image classification. Particularly, the proposed framework obtains 97.5% top - 1 image classification accuracy for the ImageNet dataset for 5 classes and 93.3% for 10 classes.**

***Keywords*-Graph embedding, image classification, transfer learning, distortion-free graph embedding, SVM-Random forest**

I. Introduction

Image classification based on graph embedding has been actively studied in many frameworks, e.g., [1]–[3]. Traditionally, these approaches represent images using hand-crafted features and construct graphs where vertices show features and edges encode relations between the features. The approaches then perform graph embedding into some geometric space such that similar graphs are located nearby while dissimilar graphs are placed further away.

Since the successful use of AlexNet [4], [5] for image classification, deep neural networks (DNNs) such as VGG [6], Inception [7], ResNet [8], DenseNet [9] have become the dominating approach for this task. Moreover, DNNs have shown their ability to learn more representative and discriminative features for image classification. Consequently, graph embedding approaches utilize deep features as opposed to hand-crafted features for graph construction.

In this paper, we propose a novel image classification using a distortion-free graph embedding with deep features. Specifically, after extracting deep features from images, we construct a complete graph such that its vertices represent features and its edges show the distances between the corresponding features. We then perform a distortion-free graph embedding under ℓ_∞ to represent the input graph as a set of points in the geometric space. Finally, we use KNN-Random Forest to perform the image classification. Experimental evaluation of the proposed method including a comparison with the previous graph embedding framework demonstrates its effectiveness.

Although the distortion-free graph embedding under ℓ_∞ using hand-crafted features has been successfully used before [3], it needs to deal with a number of problems. Namely, the previous work needs to equalize the number of features extracted from different images to make sure the embedded points are in the same dimensions. Since the proposed paper uses deep features to perform the embedding, it does not suffer from this problem. Moreover, the previous approach orders the features based on their local neighboring relations to ensure the robustness of the embedding approach. In contrast, the proposed framework solves this problem simply by using the feature ordering in the last fully connected layer of DNNs, allowing us to skip a costly and error-prone step in the embedding.

The rest of the paper is organized as follows. We review the related work in Section II and describe the proposed work in Section III. We then present the experimental evaluation of our approach and compare it against the previous framework in Section IV. We conclude the paper and draw future direction in Section 5.

II. Related work

The availability of large image datasets like imageNet [10] and the growth of the computational power of GPUs gave us the possibility to use deep learning techniques for image classification. It has been shown that this approach is superior to traditional methods [4], [11]–[15]. Deep learning techniques perform great learning the discriminative representation of images in an end-to-end fashion. Furthermore, it is possible to get deep learning features of the best performing image classification models and apply them to some other related problems [16], [17]. This technique is known as transfer learning where a pre-trained model is used.

Extracting discriminative and representative features has been a fundamental task in computer vision for decades.

978-1-7281-9891-0/20/$31.00 ©2020 IEEE
DOI 10.1109/CRV50864.2020.00013

One of the popular ways of feature extraction is using Scale-invariant feature transform (SIFT), which has been successfully applied to a number of problems, e.g., object recognition, panorama stitching, and 3D modeling. A deep learning architecture such as CNN can be used as an automatic feature extractor by using a squashing function and encoding [18].

The majority of problems in different real-world applications could be transferred to a graph-based problem. Graphs are applied in different domains like chemistry, social networks, molecular biology and computer networks due to its ability to capture hierarchical feature relations [13] and its invariance to viewpoint changes. Analyzing graphs leads to various methods of graph representations. One of the possible approaches of graph representation is to convert a graph or graph nodes into a vector space. This type of conversion acquired popularity in the research community due to the prevalence of graph representations. Depending on the characteristics such as complexity and dimensionality of the embedded space, graph embedding techniques are classified into the three main categories [19]: (1) Factorization-based, (2) Random Walk-based, and (3) Deep learning-based. Cauchy graph embedding [20], DeepWalk [21], Structural deep network embedding (SDNE) [22] are the representatives of each category, respectively. After applying one of the appropriate embedding approaches, different methods for obtaining the similarity or finding the correspondences of the embedded points are employed. For example, it is possible to use Earth Mover's Distance (EMD) [23], which is a linear optimization approach, to compute the matching between two vector spaces. This approach has been successfully used in several applications before [3]. Overall, EMD finds an optimal matching between a pair of nodes by computing the minimum amount of work required to transform one distribution into the other. Another popular family of approaches for graph embedding is based on the concept called Caterpillar decomposition [3], [24], which is the set of edge-disjoint root leaf paths. The embedding technique used in [24] introduces distortion, i.e., the distances in the embedded space are not exactly equal to those in the graph space. However, [3] uses a distortion-free graph embedding under ℓ_∞ and obtains better image classification scores. Although powerful, this approach needs to equalize the size of the input graphs and orders the vectors to properly apply the embedding. However, these steps are both error-prone and limits the applicability of the technique in practice. By using deep learning features, the proposed approach does not suffer from these problems. The experimental evaluation of this paper includes a comparison with [3].

III. Proposed Framework

Our framework consists of 6 main steps. 1) Fine-tuning of a pre-trained deep learning model for the dataset used in the proposed framework, 2) Extracting deep features for each image and them in descending order by their stability value computed in the deep learning model, 3) Creating a complete graph such that each node represents a feature and the weight of each edge reflects the absolute difference between the corresponding feature values, 4) Embedding each graph into a geometric space under ℓ_∞ without distortion, 5) Applying the hybrid algorithm KNN-Random forest for image classification, and 6) Generating the classification output.

Table I
THE ARCHITECTURE OF A MODEL FOR THE FEATURE EXTRACTION

Name	Type	Output Shape	# Params
Input	input	224x224x3	0
block1_conv1	convolution	224x224x64	1792
block1_conv2	convolution	224x224x64	36928
block1_pool	max pooling	112x112x64	0
block2_conv1	convolution	112x112x128	73856
block2_conv2	convolution	112x112x128	147584
block2_pool	max pooling	56x56x128	0
block3_conv1	convolution	56x56x256	295168
block3_conv2	convolution	56x56x256	590080
block3_conv3	convolution	56x56x256	590080
block3_conv4	convolution	56x56x256	590080
block3_pool	max pooling	28x28x256	0
block4_conv1	convolution	28x28x512	1180160
block4_conv2	convolution	28x28x512	2359808
block4_conv3	convolution	28x28x512	2359808
block4_conv4	convolution	28x28x512	2359808
block4_pool	max pooling	14x14x512	0
block5_conv1	convolution	14x14x512	2359808
block5_conv2	convolution	14x14x512	2359808
block5_conv3	convolution	14x14x512	2359808
block5_conv4	convolution	14x14x512	2359808
block5_pool	max pooling	7x7x512	0
flatten_1	Flatten	25088	0
dense_1	fully connected	200	5017800
dropout_1	dropout	200	0
dense_2	fully connected	150	30150
dropout_2	dropout	150	0
dense_3	fully connected	10	1510

We have performed our large-scale experiment using the VGG19 model for ImageNet dataset with a subset of its classes. The features extracted without a fine-tuning of VGG19 are not a good set of representative features since the original VGG19 model has been trained on 1000 classes. Therefore, the transfer learning technique has been applied. Furthermore, constructing a graph with this simplified version is computationally less expensive. In our case, we selectively train some of the last layers not just only replace the final layer. This allows us to represent an image with a fewer number of deep features compared to VGG19. To be more precise, we have frozen the last two layers of VGG19 and added layers with 200 and 150 nodes respectively (Table 1). This variation significantly helped from the computational standpoint because we do not need to store all 4096 VGG19 deep features but only 150 features for properly representing an image. After constructing a complete graph whose nodes represent deep features and whose edges reflect the distance

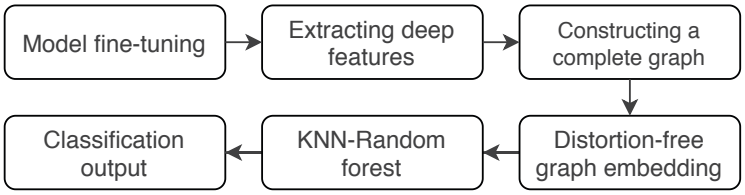

Figure 1. Overview of the proposed framework

between the corresponding features such that the distance is calculated using the Chebyshev distance, we proceed with the distortion-free graph embedding.

In particular, after obtaining 150 deep features from an image, we sort them in descending order based on their values and construct a fully connected graph. The distance between nodes is computed as the absolute difference between the features. This distance is also known as Chebyshev or chessboard distance, i.e., the distance between points X and Y is computed as

$$D_{Chebyshev}(x, y) := \max_i |x_i - y_i| \tag{1}$$

where i is the index of the corresponding coordinate. This equals the limit of the L_p metric.

$$\lim_{p\to\infty} \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{2}$$

which is also known as the ℓ_∞ metric.

A. Distortion-free graph embedding

The distortion free graph embedding under ℓ_∞ consists of several steps. Let $G = (V, E)$ be an input graph and let $V = \{v_0, v_1, v_2, v_3, v_4\}$ be its set of nodes. An embedding for a node in this graph is the set $\Omega = \{d_0, d_1, d_2, d_3, d_4\}$ where d_i is the shortest distance to the corresponding node in the graph. For instance, the vector representation for the v_0 in Fig. 2 would be $\{0.0, 2.0, 3.5, 2.0, 1.0\}$ where each element in the set is the shortest distance to a corresponding node. In the same way, we compute and get the embedding for the v_3 which gives $\{2.0, 2.5, 4.0, 0.0, 1.5\}$. In order to find an embedding for the entire graph, we find the coordinates for every node.

Feature ordering is one of the most important parts of the embedding process since this embedding is very sensitive to the order of the features. The previous work [25] order features by their relative position with respect to their neighbors, which is both costly and error-prone. In the proposed framework, we simply use the values as computed in the deep learning model. This ensures the stability of our ordering process. In addition, since there is always the same number of features obtained for an input image, we do not deal with the problem of equalizing features unlike the previous work.

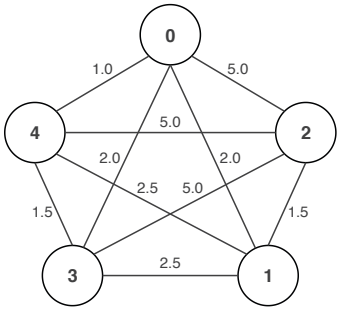

Figure 2. A sample connected graph with edge weights

Inspired by the idea of a hybrid algorithm for classification tasks, SVM-KNN combination has been successfully used as an image classifier [26]. KNN suffers from the problem of high variance, while SVM is computationally expensive. It has been shown that SVM-KNN [26] applied large multiclass datasets outperform both KNN and SVM individually. The main principle of this algorithm is to use the hybrid SVM-KNN in the following way:

- Compute the distance from the query to all other training images.
- If all of the K neighbors have the same labels, label the query accordingly.
- If not, apply the multiclass SVM after converting the distance matrix to a kernel matrix.
- Use SVM to get the query label.

Using a similar idea, we employ KNN-Random forest as the image classifier in this paper. Here, random forest is used as a replacement for SVM. A crucial reason for this replacement is due to the computational requirement of SVM especially for large datasets with a number of classes.

IV. Experiments

ImageNet is an image dataset, which was organized according to the "WordNet" hierarchy [4]. All of the meaningful concepts in Wordnet can be described with multiple words or by a word called "synsets". The total number of synsets in WordNet is more than 100000 where 80000+ of them are nouns.

Each class is represented with 1000-1500 images per category. To increase the number of images in our dataset, we have used data augmentation, which artificially expands

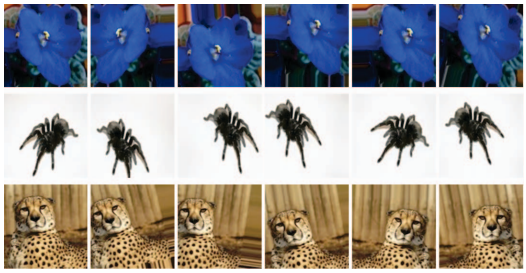

Figure 3. Augmentation examples

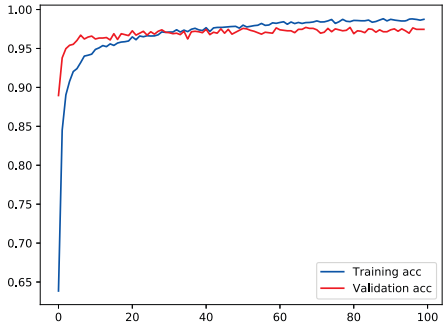

Figure 4. Training and validation accuracies for a subset of ImageNet with 5 classes

the size of a training and testing datasets by creating modified versions of all images in the dataset. This technique improves the ability to generalize for a model by feeding new variations of images. Particularly, we use a rotation of 20 degrees and the horizontal flip for each image, creating additional 5 variations per image in the dataset. Figure 3 shows this data augmentation where the first image in each row is the original image whereas the rest presents the transformed images for the classes "violet", "tarantula", and "cheetah" respectively. We use a subset of ImageNet such that the total number of images is 71326. The split proportion of the training and testing is 80% to 20% with 57061 images used for training and the rest 14265 images for testing.

According to the results, the proposed framework based on KNN-Random Forest with K=3 obtained 97.5% image classification accuracy on a subset of ImageNet with 5 classes and 93.3% with 10 classes. Figure 4 shows the training the validation accuracies of the proposed method for ImageNet with 10 classes.

For comparison, we have implemented the previous work [24], which also employs distortion-free graph embedding under ℓ_∞. Although this approach has originally been proposed with hand-crafted features, we have used the same deep features in order to fairly compare it with the proposed work. Since this method takes a tree as input, we obtain the minimum spanning tree (MST) from the fully connected graph. In particular, Kruskal's algorithm using a disjoint-set data structure has been applied. Once MST is constructed, the approach achieves embedding through the concept called Caterpillar decomposition, which captures the topological structure of the input tree as a collection of edge-disjoint root-leaf paths. One of the most important observations in this method is the embedding may give result in different dimensions for different trees. This problem potentially can be solved by dimensionality reduction techniques such as using the Principal Component Analysis [27]. Since such techniques introduces distortion, we have padded lower dimensional embeddings with zeroes to bring them up to higher dimensions. As a result, this alternative embedding technique has yielded 94% for 5 classes and 88% for 10 classes. Overall, the results demonstrate the improved image classification accuracy offered by the proposed framework. Table 4 summarizes these scores.

We should note our results are based on top-1 accuracy rather than top-5 accuracies mostly reported by the state-of-the-art methods for ImageNet. Recall that the best performing model based on a self-training method (EfficientNet-L2 [28]) achieved 87.4% of the top - 1 accuracy on the entire ImageNet. Although we have used a subset of the dataset, the results still indicate the important potential of the proposed work.

Table II
EXPERIMENTAL RESULTS OF CLASSIFICATION WITH 2 DIFFERENT EMBEDDING TECHNIQUES

Embedding technique	5 classes	10 classes
Caterpillar decomposition	94%	88%
Distortion-free full graph embedding	97.5%	93.3%

V. CONCLUSION

Graph embedding techniques have been employed by several different frameworks for a number of problems, such as image classification, feature correspondence, and image indexing. In this paper, we have proposed an image classification framework based on distortion-free graph embedding with deep features. Although such distortion-free graph embedding has been proposed with hand-crafted features before, the way that we apply this embedding using deep features overcoming some problems faced by the alternative technique is novel. We have shown the effectiveness of the proposed framework in a subset of ImageNet. However, our future goal is to perform a more comprehensive evaluation in a larger dataset and compare it with more alternative methods.

ACKNOWLEDGEMENTS

This research was funded under the Nazarbayev University faculty development grant "Forming Reliable Feature Correspondences and Distortion-free Graph Embedding

with Deep Learning". Project PI - M.F. Demirci, Grant# 110119FD4530.

REFERENCES

[1] L. Shi, L. Zhang, J. Yang, L. Zhang, and P. Li, "Supervised graph embedding for polarimetric sar image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 2, pp. 216–220, 2012.

[2] Z. Xue, P. Du, J. Li, and H. Su, "Simultaneous sparse graph embedding for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 11, pp. 6114–6133, 2015.

[3] M. F. Demirci and S. Kacka, "Object recognition by distortion-free graph embedding and random forest," in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*. IEEE, 2016, pp. 17–23.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[5] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[7] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[12] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647–655.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition international conference on learning representations (iclr)," 2015.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[16] M. Vakalopoulou, K. Karantzalos, N. Komodakis, and N. Paragios, "Building detection in very high resolution multispectral data with deep learning features," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2015, pp. 1873–1876.

[17] J.-C. Chen, V. M. Patel, and R. Chellappa, "Unconstrained face verification using deep cnn features," in *2016 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2016, pp. 1–9.

[18] F. Shaheen, B. Verma, and M. Asafuddoula, "Impact of automatic feature extraction in deep learning architecture," in *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2016, pp. 1–8.

[19] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.

[20] D. Luo, F. Nie, H. Huang, and C. H. Ding, "Cauchy graph embedding," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 553–560.

[21] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.

[22] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 1225–1234.

[23] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, no. 2, pp. 99–121, 2000.

[24] M. F. Demirci, Y. Osmanlioglu, A. Shokoufandeh, and S. Dickinson, "Efficient many-to-many feature matching under the l1 norm," *Computer Vision and Image Understanding*, vol. 115, no. 7, pp. 976–983, 2011.

[25] M. F. Demirci, A. Shokoufandeh, Y. Keselman, L. Bretzner, and S. Dickinson, "Object recognition as many-to-many feature matching," *International Journal of Computer Vision*, vol. 69, no. 2, pp. 203–222, 2006.

[26] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "Svm-knn: Discriminative nearest neighbor classification for visual category recognition," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 2126–2136.

[27] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[28] Q. Xie, E. Hovy, M.-T. Luong, and Q. V. Le, "Self-training with noisy student improves imagenet classification," *arXiv preprint arXiv:1911.04252*, 2019.