Graph-Based CNN for Human Action Recognition from 3D Pose

Michael Edwards

Xianghua Xie
csvision.swan.ac.uk

Department of Computer Science
Swansea University
Swansea, SA2 8PP
United Kingdom

Abstract

Deep learning has shown increasingly promising performance in the pattern recognition field in recent years, becoming a prominent staple in image classification problems since the introduction of the convolutional neural network. Such CNN models work well in the image domain due to the spatially regular structure of the 2D and 3D grid, but not all domain exhibit such a regular spatial structure. In order to retain the underlying spatial information within the domain application, this study presents operators for graph-based convolution and pooling, utilizing graph based signal processing methods to define common deep learning operators, such as convolution and pooling, on a graph representation of the spatial human skeleton domain. The proposed method avoids unnecessary assumptions of spatial relationships between hand-crafted features, and evaluation shows strong sequence classification rates that exceeds 93%.

1 Introduction

Deep learning has been a prominent feature in data mining and pattern recognition in recent years, especially in problems such as classification and detection. Fully connected neural networks have shown promising usage in feature space learning in domains including text document analysis and genome characterization [45], with numerous architectures being designed that are able to self-tune features to the problem under investigation [18, 36]. By providing low level or raw input features, deep learning methods have been shown to learn high level descriptive features for various structures within the data [34, 42]. Such methods exhibit strong performances in various testing scenarios [18] and show promise for further data mining problems [25].

Convolutional Neural Networks (CNNs) expanded upon the concept of neural networks, learning localized features by convolving kernel filters with the input space to generate output feature maps [19]. With localization of features came a great increase in the ability of networks to learn descriptors in image mining problems [21, 22], and CNNs have shown promising applications in a wide range of image based data learning problems; including digit classification [7], face detection [20], and classification on a large number of classes [28]. CNN architectures presented two key operators, convolution and pooling, to learn spatially localised features.

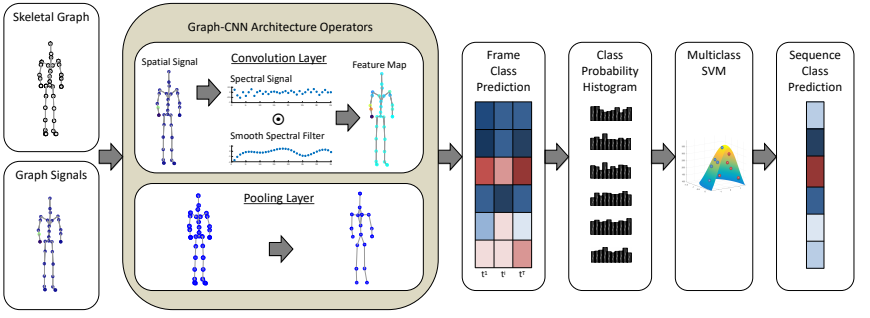


Figure 1: Graph based Convolutional Neural Network components. The GCNN is designed from an architecture of graph convolution and pooling operator layers. Output prediction probabilities from classification on frames $t^{1:T}$ are histogram binned and passed into a multiclass SVM for sequence classification.

One limitation of the CNN convolution operator is the assumption of regular spatial topology in the problem domain. This is readily defined in the image domain, with the 2D and 3D grids providing a traversable spatial domain upon which to define locally receptive fields [17, 39]. This assumption is not so apparent when considering problem domains that do not reside on a regular spatial grid, and as such it is a non-trivial problem to define a convolutional operator that is able to traverse localized regions of the input space. Current methods of deep learning recognition on such an irregular input space include using standard neural networks [43], embedding features on the regular grid to allow standard CNN convolution [15, 16], or identifying local manifold patches for geodesic convolutions [24]. These methods either ignore important spatial information of the domain, or forcefully define arbitrary spatial relationships that may not be appropriate to the domain.

In this study, we utilize graph-based signal processing techniques to generate a GCNN architecture on irregular domain problems; presenting GCNN convolution and pooling operators for use in Human Action Recognition (HAR) learning systems. Recent study has shown that graph-based signal processing techniques can learn on irregular domains present in a wide range of applications [8, 10, 14]. The concept of employing the graph Laplacian to undertake signal processing based kernel learning on geometrically irregular space was first introduced in [8], while [14] goes on to explore use of smooth filters to identify localized regions in the spatial domain. The presented GCNN operators are utilized to construct deep learning architectures for problem domains beyond image processing and the regular CNNs.

We believe that this study shows the first usage of the GCNN architecture for the HAR from 3D pose problem, implementing deep learning in the natural spatial domain of the 3D pose. The proposed GCNN avoids hand-tuning features and the spatial embedding utilized by current methods to adapt the 3D pose information into the regular CNN framework. By using very low level features of motion the network is able to learn spatial relationships on the irregular domain of the graph by the proposed convolution and pooling operations.

The rest of the paper is as follows. Section 2 describes GCNN architecture, providing convolution and pooling operators in the graph domain by use of graph based signal-processing. A domain specific application is then presented in the context of human action recognition in Section 3. Conclusions are then drawn in Section 4.

2 Methods

The formulation of CNN convolution becomes problematic when considering a domain in which there is no regular spatial structure, notably the application of operators translation across the space and definition of a localized vertex neighborhood. One solution is to utilize the analogy between multiplication in the graph spectral space and convolution in the spatial domain. Localized feature maps can therefore be computed on an irregular domain graph by graph signal processing techniques of graph Fourier transforms and spectral filtering [2, 33]. The graph forms a carrier for an observed graph signal and holds an underlying knowledge about the spatial relationship between vertices [33]. By combining graph signal processing operators and deep learning architecture design it is possible to learn on irregularly spaced domains upon which conventional CNNs would be unable to convolve a regular kernel.

Below we describe the construction of GCNN operators that are used to develop a deep learning GCNN network on the domain of HAR from 3D pose features. These methods are then implemented and evaluated for sequence-wise HAR classification in Section 3. See Figure 1 for an overview of the general proposed GCNN architecture components.

2.1 Graph Representation of the Irregular Domain

A graph $G = \{V, W\}$ contains N vertices V and a weight matrix W of the non-negative, undirected, non-self-looping edges between two connected vertices v_n and v_m. G is an edge weighted graph, with no weightings associated to its vertices. Edge weighting and connectivity can therefore be described by its unnormalized graph Laplacian matrix L, defined as $L = D - W$, where D is a diagonal matrix $d_{n,n} = \sum_{n=1}^{N} a_n$ containing the sum of all adjacencies $a_{n,1:N}$ for a vertex n from the binary node adjacency matrix A. It is possible to use a normalized L, but for this study we take the unnormalized form of L, as similar performance is observed when utilizing graph signal processing operators [33]. Generating a graph can be non-trivial and is domain specific, with study into how to generate edge weightings between vertices still a hot topic of discussion [12, 33, 44].

2.2 Graph Signals

Given G with N nodes, an observed data sample is a signal $f \in \mathbb{R}^N$ residing on G, where f_n is the signal amplitude at vertex v_n. For an I-channeled observation f becomes a $N \times I$ matrix, where each $f_i \in \mathbb{R}^N$ is a vector of features associated to vertices for the ith channel. These input channels can then be subjected to localized feature learning via the graph convolutions and deep learning architectures of the proposed GCNN. By having the graph represent the underlying relationships between the vertices we are able to define a fixed spatial relationship of inputs, irrespective of the value they hold.

2.3 Convolution on Graph

Due to the irregular topology exhibited by the domain of interest we cannot directly use the regular convolutional operator defined in standard CNNs. Instead we learn localized features via the convolution theorem and the spectral graph form of f [2]. To project f into the frequency domain, the Laplacian L is decomposed into a full matrix of orthonormal eigenvectors $U = \{u_{i=1\ldots N}\}$, where each eigenvector is a column u_i in U, and its vector of associated eigenvalues $\lambda_{i=1\ldots N}$. Such eigen decomposition describes the frequencies present

on the graph structure based on the neighborhoods defined by the Laplacian matrix, allowing a given graph signal f to be represented in the spectral frequency domain by a Fourier transform on the graph eigenvectors. To obtain the $\mathbb{R}^N$ spectral form of f_i, we define the Graph Fourier Transform as a matrix multiplication between the eigenvector matrix U and the signal $\tilde{f}_i = U^T f_i$, with the inverse given as $f_i = U\tilde{f}_i$, where U^T is the transpose of the eigenvector matrix.

Using the convolution theorem, a convolutional operator in the vertex domain can be composed as elementwise multiplication in the Fourier domain of the Laplacian operator L [2]. Given $\tilde{f} \in \mathbb{R}^N$ and a spectral multiplier filter $k \in \mathbb{R}^N$, the spatial domain feature map y is given by $y = U(\tilde{f} \odot k)$. For multiple input channels and multiple output feature maps we can summate over the convolutions of individual input channels for each output map:

$$y_{s,o} = U(\sum_{i=1}^{I} U^T(f_{s,i}) \odot k_{i,o}) \tag{1}$$

where I is the number of input channels associated with f, s a given observation sample, and o indexes an output feature map from O desired output maps. The $\odot$ describes the elementwise multiplication of $\mathbb{R}^N$ spectral signal $\hat{f}_{s,i}$ and $\mathbb{R}^N$ spectral multiplier $\hat{k}_{i,o}$.

2.4 Pooling on Graph with Kron's Reduction

Regular CNN architecture often pools input feature channels by striding a receptive cell across the spatial domain, evaluating an appropriate max or mean operator to produce a reduced resolution map. The pooling operator eases the scaling ability of architectures and generalizes feature maps by resolution compression [1]. This pooling operator maintains the spatial regularity of the domain, returning a Euclidean grid feature map. During graph based convolutions there is no reduction in feature map size, due to the elementwise multiplication of the spectral filter with the spectral input signal. As such, each layer of a GCNN would possess a graph with $\mathbb{R}^N$ vertices and the increasing output maps would quickly succumb to scaling inefficiencies. We can however formulate a similar pooling operator for our GCNN architecture, reducing the number of vertices in the graph and handling the graph signal appropriately. Pooled graphs will also benefit from reduced complexity in convolution operations, due to the reduced size of the eigenvector matrix for $\hat{G}$. As observations in a batch are treated as graph signals on a single graph, GCNN architectures store a single copy of G and U for each graph resolution, casting the observations onto the correct graph for that layer. This allows for the pre-computation of the different graphs required for the entire architecture, given that the number of pooling layers are known in advance. Pooling $G = \{V, W\}$ to $\hat{G} = \{\hat{V}, \hat{W}\}$ is by no means trivial; with extensive literature exploring possible methods of removing, merging, or clustering vertices. [23, 30, 31]. Common methods for selecting $\hat{V}$ are to either select a subset of V, [32], or generate new nodes $\hat{V}$ from aggregation of related nodes in the spatial or spectral domains, [30].

For this domain problem we utilize Kron's reduction [9]. Kron's reduction provides a means to reconstruct the reduced node weight matrix $\hat{W}$, via the removal of discarded vertices from the rows and columns of the original graph Laplacian L. Selected vertices are used to construct the coarsened $\hat{G}$. Kron's reduction has the effect of increasing the number of edge connections present in the graph, and as such it is often necessary to sparsify the connectivity in the graph by way of spectral sparsification [32, 35]. New edge weights are accumulated into the new subgraph's weight matrix for the coarser graph layer based on a prior probability

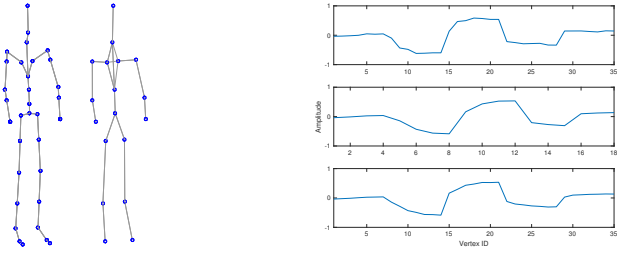
distribution. With a coarser graph structure, $\hat{G}$, it is necessary to downsample the graph signal $f_{1:N}$ into a new signal $\hat{f}_{1:\hat{N}}$ that is able to reside on $\hat{G}$. Kron's pyramid utilizes a linear application of Green's functions, derived from the Laplacian, to interpolate the signal about a given vertex v_n in the spatial domain [32]. This allows us to project our samples from fine to coarse resolutions during forward passes through the network, and from coarse to fine scale during the backpropagation of errors.

3 Domain Application: Recogntion of Human Action from 3D Pose

This study focuses on the use of GCNN within Human Action Recognition of 3D skeletal pose, with classification on the Berkeley Multimodal Human Action Database (MHAD) dataset [27]. By converting the human skeletal model to a graph based representation, we are able to utilize our GCNN method without arbitrarily defining a set of hand-crafted high level features, or projecting the data into a regular space just to suit standard CNNs.

3.1 Implementation

MHAD contains 11 action classes, performed by 12 subjects, and captured via an array of modalities. We utilize the 3D motion capture information, omitting appearance information in this instance. We normalize the data on a sequence-by-sequence basis in both orientation to camera and scale, as per the normalizing algorithm presented by [4]. Due to the high frame capture rate, we sub-sample the sequences down from 480Hz to 30Hz, bringing it in line with commercial pose capturing sensors such as the Microsoft Kinect and Kinect V2. The problem of human pose has a well-defined structure of connectivity to formulate into a graph. Tracked 3D points in Motion Capture (MoCap) data constitute the graph vertices on the graph of the human body, and the adjacency between these points (bones) can be defined by the human skeleton in binary adjacency matrix A. From this prior knowledge of the domain we can define connected vertices for the human skeleton as in Figure 2. Given all training observations and the joint adjacency matrix A, we can generate weight matrix W for the edge between adjacent vertices n and m across all observations, weighting the bone edges by the inverse of the mean Euclidean distance between the end joints of a given bone. By adding the adjacency matrix into the weight matrix, we are able to define weighted edges for data points that occupy the same physical space. Such phenomena can be common in 3D pose data for smaller digits such as fingers and toes where confidence of tracking is low. The final form of W provides zeros for non-edge connection, ones for an edge which occupies the same XYZ locations on its two end points, and a value larger than one for all edges with a distance based weight. Due to their prominent use in pose based action recognition, we wish to learn on low level joint motion features from each frame of the observation, [8, 11, 41]. We extract the XYZ coordinates, along with multi-scale motion features of velocity and acceleration for all tracked markers, returning an $V \times I \times X$ matrix of X frames with $I = 123$ channel graph signals residing on $V = 35$ vertices. We extract the features for each of the 3 spatial dimensions X, Y, and Z attributed to $v_n^{\mathrm{X,Y,Z}}$, calculating the velocity as $vel(v_n^{\mathrm{X}}) = \frac{\Delta v_n^{\mathrm{X}}}{\Delta t}$. Velocity is calculated in relation to directional vectors of upper back to left shoulder, upper back to right shoulder, horizontal shoulder to shoulder, and vertical upper back to lower back. Acceleration is then given as $acc(v_n^{\mathrm{X}}) = \frac{\Delta vel(v_n^{\mathrm{X}})}{\Delta t}$, where t defines the time step and v_n^{X}



(a) (b)

Figure 2: Human skeletal graph for the MHAD MoCap data. a) Connective adjacency of the MoCap markers and Kron's reduction coarsened human graph. b) Graph signal pooling: from top to bottom: original signal, pooled signal, upsampled signal. Detailed in Section 2.4.

is a given spatial dimension X, Y and Z of the tracked point v_n. Velocities are extracted for sub-second frame steps, calculating motion on the scales of $\frac{1}{30}^{\text{th}}$, $\frac{1}{10}^{\text{th}}$, $\frac{1}{5}^{\text{th}}$, $\frac{1}{4}^{\text{th}}$, and $\frac{1}{2}^{\text{th}}$ of a second (rounded up to the nearest frame), resulting in short-scale frame motion information.

3.2 Evaluation

Evaluation on the MHAD dataset has been carried out in several ways by previous studies. The initial paper reports a 7vs5 approach, training on the observations of subjects 1 to 7, and testing on subjects 8 to 12 [27]. The overall HAR GCNN sequence classifier is as follows. First a GCNN is trained to predict probability of a frame belonging to one of the possible classes. A histogram is taken of the returned frame-wise class probabilities for each sequence. This compresses the temporal dimension of the observations into a fixed length feature vector. This vector is then used to train a multi-class SVM to classify whole sequences into an action class. For testing sequences, each frame is fed forward through the GCNN and their classes probabilities are then compressed via histogram binning to fit into the pre-trained SVM. The SVM returns a prediction on the class label for the entire sequence.

The architecture of the graph CNN is defined as $C^{20} - P - C^{50} - R - F$; where C^{κ} defines a graph convolutional layer with 5 knots and κ output feature maps, P defines a graph coarsening, R defines a rectified linear unit layer, and finally F describes fully connected layers providing output class predictions. Graph pooling was achieved via Kron's reduction and spectral sparsification. The two human skeleton graphs used in the architecture, and signal pooling can be seen in Figure 2. After GCNN training, we performed a forward pass with the training data and a histogram was taken of the output predictions from the fully connected neural network layer, returning a fixed length feature vector for each sequence. These sequences were then used to train a multi-class SVM classifier. The test set was then fed forward through the GCNN in the same manner, and histogram representation of the testing sequence probabilities were then classified using the pre-trained SVM. We report on the final sequence-wise classification accuracy for the original 7vs5.

3.3 Results

The proposed Graph-CNN methods achieves a 93.82% accuracy in the 7vs5 validation scenario, improving over the baselines reported by [27], and also over newer methods presented by [6]. [27] obtains accuracy of 74.82%, 75.55%, and 79.93% for a 1 Nearest Neighbour, 3

Nearest Neighbour and K-SVM approach respectively. [6] obtains 87.83% and 89.85% utilising understanding actions and execution styles via bi-linear modelling. A 100% accuracy reported by [5] is an obvious issue; despite this, our proposed GCNN provides a benefit over [6] and [5] in that we utilize very low level features, in comparison to the large number of hand-crafted temporal features used in the current state of the art methods.

In all of the closest state of the art results the use of hand-crafted features is evident. Although these features can provide strong performances on a given dataset, it is often difficult to apply them on a new HAR scenario due to their selection of informative joints and feature extractors. Using heavily hand-crafted features are at odds with the self-learning feature extractors of common deep learning methods such as CNNs, autoencoders, and the proposed GCNN. GCNNs are able to optimizing towards informative features, obtaining an understanding of the initial observations based on very low level or even raw data input. We are able to train GCNN with very low level motion and spatial information regarding each of the joints on the human skeleton, and from here the algorithm is able to learn generalized features for frame-wise classification.

Overall the proposed GCNN has shown strong performance in the domain of 3D pose based HAR. The graph convolution operator presented is able to generate feature maps on the spatially irregular graph of the human skeleton, acting as a learnable feature extractor when trained within a deep learning framework. The graph coarsening operator allows us to reduce the graph resolution in order to generalize feature maps and reduce complexity. We have shown favorable classification accuracies on a public HAR dataset, especially given that the rival methods all utilize sets of user tuned features.

4 Conclusion

This study has proposed a method for the end-to-end mining of localized features in domains with irregular geometry. The combination of graph signal processing techniques and deep learning architecture design has allowed for features to be learnt on low level data in an end-to-end fashion. The local features are learnt by the use of spectral domain convolution of graph signals and spectral multipliers, in architecture similar to that seen in regular usage within standard CNNs. Convolutions are performed in the spectral domain of the graph Laplacian and allow for the learning of spatially localized features via the gradient calculations provided. Results are provided on the domain of HAR, although the scope for further application is much wider. Evaluation on HAR in a range of cross validation scenarios shows the ability of GCNN to learn localized feature maps for frame-wise classification.

References

[1] Y-Lan Boureau, Jean Ponce, and Yann Lecun. A theoretical analysis of feature pooling in visual recognition. In *Proc. Int. Conf. Mach. Learning*, 2010.

[2] Ronald Bracewell. *The Fourier Transform & Its Applications*. McGraw, 1999.

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *CoRR*, abs/1312.6203, 2013.

[4] Alexandros Andre Chaaraoui, José Ramón Padilla-López, Pau Climent-Pérez, and Francisco Flórez-Revuelta. Evolutionary joint selection to improve human action

recognition with RGB-D devices. *Expert Systems with Applications*, 41(3):786–794, 2014. ISSN 09574174. doi: 10.1016/j.eswa.2013.08.009.

[5] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal. Bio-inspired dynamic 3d discriminative skeletal features for human action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 471–478, 2013. doi: 10.1109/CVPRW.2013.153.

[6] Muhammad Shahzad Cheema, Abdalrahman Eweiwi, and Christian Bauckhage. Human activity recognition by separating style and content. *Pattern Recognition Letters*, 50:130 – 138, 2014. doi: 10.1016/j.patrec.2013.09.024.

[7] Dan C. Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *CoRR*, abs/1202.2745, 2012.

[8] Jingjing Deng, Xianghua Xie, and Ben Daubney. A bag of words approach to subject specific 3d human pose interaction classification with random decision forests. *Graphical Models*, 76(3):162 – 171, 2014. doi: http://dx.doi.org/10.1016/j.gmod.2013.10.006.

[9] F. Dorfler and F. Bullo. Kron reduction of graphs with applications to electrical networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(1):150–163, 2013. doi: 10.1109/TCSI.2012.2215780.

[10] M. Edwards and X. Xie. Graph based convolutional neural network. In *Proc. British Mach. Vis. Conf.*, 2016.

[11] Michael Edwards, Jingjing Deng, and Xianghua Xie. From pose to activity: Surveying datasets and introducing {CONVERSE}. *Comp. Vis. Image Underst.*, 144:73 – 105, 2016. doi: http://dx.doi.org/10.1016/j.cviu.2015.10.010.

[12] Leo Grady and Jonathan R. Polimeni. *Discrete Calculus - Applied Analysis on Graphs for Computational Science.* Springer, 2010.

[13] David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129 – 150, 2011.

[14] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

[15] E. P. Ijjina and C. K. Mohan. Human action recognition based on mocap information using convolution neural networks. In *Proc. Int. Conf. Mach. Learning and Applications*, pages 159–164, 2014. doi: 10.1109/ICMLA.2014.30.

[16] E. P. Ijjina and C. K. Mohan. Human action recognition based on motion capture information using fuzzy convolution neural networks. In *International Conference on Advances in Pattern Recognition*, pages 1–6, 2015. doi: 10.1109/ICAPR.2015.7050706.

[17] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 1097–1105. Curran Associates, Inc., 2012.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec.*, pages 5325–5334, 2015.

[21] Y. Li, L. Liu, C. Shen, and A. van den Hengel. Mid-level deep pattern mining. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec.*, pages 971–980, 2015. doi: 10.1109/CVPR.2015.7298699.

[22] Yao Li, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. Mining mid-level visual patterns with deep CNN activations. *CoRR*, abs/1506.06343, 2015.

[23] P. Liu, X. Wang, and Y. Gu. Graph signal coarsening: Dimensionality reduction in irregular domain. In *IEEE GLobal Conf. Signal and Information Processing*, pages 798–802, 2014.

[24] Jonathan Masci, Davide Boscaini, Michael M. Bronstein, and Pierre Vandergheynst. Shapenet: Convolutional neural networks on non-euclidean manifolds. *CoRR*, abs/1501.06297, 2015.

[25] Maryam M. Najafabadi, Flavio Villanustre, Taghi M. Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1–21, 2015. doi: 10.1186/s40537-014-0007-7.

[26] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *IEEE Workshop on Applications of Computer Vision*, pages 53–60, 2013. doi: 10.1109/WACV.2013.6474999.

[27] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, Rene Vidal, and Ruzena Bajcsy. Berkeley MHAD: A comprehensive Multimodal Human Action Database. *Workshop on Applications of Computer Vision*, pages 53–60, 2013. doi: 10.1109/WACV.2013.6474999.

[28] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec.*, pages 1717–1724, 2014.

[29] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proc. British Mach. Vis. Conf.*, 2015.

[30] Ilya Safro. Comparison of coarsening schemes for multilevel graph partitioning. In *Int. Conf. Learning and Intelligent Optimization*, pages 191–205. Springer-Verlag, 2009.

[31] Ilya Safro, Peter Sanders, and Christian Schulz. *Proc. Int. Symposium Experimental Algorithms*, chapter Advanced Coarsening Schemes for Graph PartitioninG, pages 369–380. Springer Berlin Heidelberg, 2012.

[32] D. I. Shuman, M. J. Faraji, and P. Vandergheynst. A multiscale pyramid transform for graph signals. *IEEE Trans. Signal Process.*, 64(8):2119–2134, 2016.

[33] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

[34] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[35] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pages 563–568, New York, NY, USA, 2008. ACM. doi: 10.1145/1374376.1374456.

[36] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec.*, pages 1–9, 2015. doi: 10.1109/CVPR.2015.7298594.

[37] S. Vantigodi and R. Venkatesh Babu. Real-time human action recognition from motion capture data. In *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, pages 1–4, 2013. doi: 10.1109/NCVPRIPG.2013.6776204.

[38] S. Vantigodi and V. B. Radhakrishnan. Action recognition from motion capture data using meta-cognitive rbf network classifier. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6, 2014. doi: 10.1109/ISSNIP.2014.6827664.

[39] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. 2015.

[40] Ashok Veeraraghavan, Anuj Srivastava, Amit K Roy-Chowdhury, and Rama Chellappa. Rate-invariant recognition of humans and their activities. *Transactions on Image Processing*, 18(6):1326–39, 2009. doi: 10.1109/TIP.2009.2017143.

[41] Angela Yao, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Does human action recognition benefit from pose estimation? In *Proc. British Mach. Vis. Conf.*, pages 67.1–67.11. BMVA Press, 2011. ISBN 1-901725-43-X.

[42] Jason Yosinski, Jeff Clune, Anh Mai Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *CoRR*, abs/1506.06579, 2015.

[43] Bo Yu and Dong hua Zhu. Combining neural networks and semantic feature space for email classification. *Knowledge-Based Systems*, 22(5):376 – 381, 2009. doi: 10.1016/j.knosys.2009.02.009.

[44] Cha Zhang, Dinei Florencio, and Philip Chou. Graph signal processing - a probabilistic framework. Technical Report MSR-TR-2015-31, 2015.

[45] Min-Ling Zhang and Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.