A Dissertation Report
on

Image Recognition Using Graph Based Convolutional Neural Network

Submitted for the partial fulfillment of the requirements for the award of the course credits for Masters of Technology

Submitted By:

Sheetal Yadav

Enrollment No.: 16535036

Under the guidance of

Dr. Biplab Banerjee

Assistant Professor

Department of Computer Science and Engineering

Indian Institute of Technology

Roorkee

May, 2018

Declaration

I declare that the work presented in this dissertation with title **”Image Recognition Using Graph Based Convolutional Neural Network ”** towards fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science & Engineering** submitted in the **Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period of **May 2017 to May 2018** under the supervision of **Dr. Biplab Banerjee**, Assistant Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, India. The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

Date: .... SHEETAL YADAV
Place: ROORKEE (16535036)
M.TECH (CSE)

Certificate

This is to certify that the statement made by the candidate is correct to the best of my Knowledge and belief.

Date: ....

Place: ....

Sign: ....

Dr. Biplab Banerjee
(Assistant Professor)
Indian Institute of Technology
Roorkee

Abstract

The success behind Deep Learning have mostly relied on Convolutional Neural Network. Convolutional Neural Networks became popular because of their efficient ability to exploit significant statistical properties of images, audio and video data which allows depicting long range interactions in the form of smaller, localized interactions. In Machine Learning, localized feature in the regular domain boosted the use of Convolutional Neural Network, with great advancement in the image processing and classification. But there exist some domains such as social networks, bio-informatics data which lack few or all of these fundamental statistical properties and considered as the high-dimensional irregular domain. Being non-trivial in the design and convolution of a kernel filter there arises an issue with the use of Convolution Neural Network within irregular spatial domain. Solution to this problem can be in two direction, where one is to represent these high dimensional irregular domains using graph and then use graph signal processing methods and theorems to execute convolution on graph structure of irregular domain to extract features maps to learnt filters. So, graph convolution and pooling operators like those for regular domain can be a solution to this problem. Other solution to this problem can go in a direction where by calculating gradients on the data input and spectral filter, can achieve deep learning of a problem of irregular spatial domain. Here we will focus on general query of how to build deep networks on non-Euclidean domains in context of spectral theory over graphs with small complexity in its learning. Importantly, the suggested method will offer almost the same constant learning and computational complexity as offered by standard CNNs and extended to any graph structures. The experiments carried on MNIST and Pascal VOC 2012 datasets depict the ability and efficiency of the proposed deep network to learn the statistical and compositional features of these high-dimensional irregular domains as represented through graphs.

Acknowledgement

Dedicated to my family and friends, for standing by me through thick and thin, without whom i would not have gotten this far. I would like to express my sincere gratitude to my advisor **Dr. Biplab Banerjee** for the continuous support of my study and research, for his patience, motivation, enthusiasm and immense knowledge. His guidance helped me in all time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my study.

I am also grateful to the Department of Computer Science and Engineering and Rethink Lab, IIT Roorkee for providing valuable resources to aid my research.

SHEETAL YADAV

Contents

List of Figures

Chapter 1

Introduction

In recent times, for classification problems the computer vision and machine learning community has seen a revival in the practice of deep learning and neural network. The neural network models have efficiently been applied to a wide variety of work ranging from pattern recognition, computer vision and modeling to NLP(Natural Language Processing). The foundation of their success and research lies on an important assumption on the properties of data, the properties like ” *stationarity*” and ”*compositionality* ” by using local statistics present in images, audio and video data. Since the properties are within regular domain and fully connected neural network are used efficiently in [17, 20] to extract the local features present across the domain of signal. ConvNets[30], analogous to deep network architecture has greatly decreased the count of parameters without compromising the capacity to extract descriptive statistics from the input data. Convolutional Neural Network [24] gave an efficient way to extract features with the help of kernel filter convolution across the spatial domain of input data. CNNs played a great role in pattern and image recognition, such as face detection[22], text analysis etc.

Convolutional Neural Networks (CNNs) have been proved an extremely efficient architecture for those problems in machine learning where to represent the data the coordinates have a grid like structure and the data has translational invariance or equi-variance. The example of such data can be imagined as defining signals on a low dimensional grid. Images [20, 21], Speech[17] and video [37] are examples which resides under this category. Translational operator on the grid defines the stationarity, metric of grid defines the locality and multi-resolution and down-sampling property of the grid defines compositionality. Yet there are many examples of datasets are there that lack these low-dimensional grid properties and hence make it difficult for the convolutional neural network to classify such high-dimensional grid. Computer vision and audio being the major research area of deep learning only depicts the special type of data described on an absolutely low dimensional graph. The complex graph which we encounters in the other domains might be of higher dimension and their fundamental statistical properties might lack stationarity, locality and compositionality.

Convolutional Neural Networks acknowledges the local features over a data domain by extracting the stationarity property of the given input signals locally. Localized convolutional kernels which are learned from the input data identifies these identical features independent of their locations spatially because they are shift-invariant kernels. User data lying on social networks, genetic data, text documents on word embeddings, or data logs on telecommunication networks are some of the important examples of datasets residing in the non-Euclidean domain or irregular domains and these datasets can be structured using graphs.

The idea of ConvNets brought up the hidden convolutional layers and pooling layers to analyze structurally localized feature maps in kernel form using set of receptive fields [12]. The input data once fed to the convolutional layer then the operator convolves kernel filters in the structural domain of the input data along with the user specified parameters such as stride and padding. In response to these filters, the convolutional layer will return the feature maps. If the input is given through multiple channels then the resulting feature map can be obtained as summation of convolution with each filters for separate channels. In pooling layer in CNN, operator is used to reduce the resolution of responses to each kernels (feature maps) in the structural dimension of input data without altering the count of feature maps. With the help of pooling, the network is able to generalize the feature map by resolution reduction and handles the increasing count of the feature maps. Average and max of cells in the input feature map are basic operations in the pooling layer[3].

So far the use of convolutions was for extracting feature maps with an assumption that the input data provide some structural locality. In many implementation of CNNs, the convolutions on the spatial domain is well presented, but problem arises when we shift to irregular domains. Defining a kernel filter for an irregular domain is non-trivial, hence identifying a way to translate such a kernel filter in the spatial domain is difficult[30].

Graphs are the common form of data representation for defining geometry of data in various applications, considering social, sensor and artificial neural networks. The edge between vertices having some weight defines how similar they are. In recent advancements in the image processing, graph -based approaches that construct graphs to connect the pixels of an image based not only on how close they are physically but also on image's noisy version[5]. There are many question that are required to be answered like- How we can process the data on irregular domains? How to extract feature maps using convolutional neural network over an irregular spatial domain? These are the questions that highlights the branch of graph signal processing. Both a discrete-time signal with **V** samples and a signal residing on graph with **V** vertices can be shown as vectors in $\mathbb{R}^V$. If the processing of graph signal is carried in similar way like the discrete-time signal, it avoids the primary dependencies stemming from the irregular spatial domain[39].

Therefore, the major challenges in graph signal processing are

1. The graph is not directly given to us by the application problem and making it difficult to decide how to construct an undirected weighted graph that can preserve the geometric structure of the latent data domain.
2. Including graph structure in the localized transforming methods.
3. Applying the priceless insights from researches of graph signal processing in regular domain to irregular domain.
4. Implementation of these localized transform, to extract feature maps from graph signals on irregular input data domain.

To solve the above challenges, graph signal processing [36] combines concepts from both spectral graph and algebraic concepts with harmonic analysis. The spectral graph theory has been proved as a tool to describe the frequency spectrum and the foundation of graph Fourier transform[4]. The *spectral* construction of graph is based on the fundamental characteristics of convolution over the Fourier transform. In $\mathbb{R}^V$, convolutions are like linear operations which are diagonalized with the Fourier basis exponent. Due to this the convolution can be widened to common type of graphs by evaluating their respective "Fourier" basis. Graph Laplacian[28], which give a harmonic analysis provided this correspondence in the graph spectral domain and it requires $O(V)$ parameters per feature maps . It also allow a highly efficient forward propagation and using Laplacian spectrum[16] for mining feature maps during gradient descent provide derivative for backward propagation of the errors.

Although the graph Fourier transform aids the working over graph signals, the computational cost required in convolutional layer for the eigenvalue decomposition and storing the Fourier basis can be prevented as per our proposed model.

The idea of generalizing convolutional neural networks is not simple because the convolution and pooling operators are only described for the regular domain grids. The concept of extending it for the irregular domains makes it challenging in both the way theoretically and practically. Defining localized graph filters responsible for the evaluation and learning are proved to be the major obstacle in this generalization.Convolution being an expensive task for irregular domain problems, there is research going on the graph convolution and pooling with data as an arbitrary graph[28].

Particularly, the proposed work will constitute the following:

1. Spectral Formulation: Using established tools defined in graph signal processing, a spectral theoretical formulation of CNN's is established on graphs.

2. Highly localized filters: The spectral filters proposed using enhancement, found to be localized strictly within a radius around the vertex.

3. Better computational complexity: Our filters performs linearly w.r.t to its input size. Generally, most of the graphs in real scenarios are highly sparse, resulting in linear complexity w.r.t input data size. In addition, this method do not require Fourier basis for its computation, thus avoiding the expensive eigenvalue decomposition used to evaluate it, and the space used for storing the basis. This is really fruitful while working with limited GPU memory. Along with the data, the proposed method needs the storage of Laplacian matrix, which is sparse in nature.

4. Efficient pooling: Like pooling of one-dimensional signals, we are applying an efficient pooling method on graph by rearranging the vertices as a binary tree structure.

5. Experimental results: Experiments proves the model as useful and computationally better and efficient both in accuracy and complexity to the earlier spectral

graph CNNs introduced in [5]. It also shows that the network model performs analogous to a standard CNNs on MNIST and Pascal VOC 2012.

Before the input data fed to our model, it requires some preprocessing as our datasets is in form of images(.png), but our Graph CNN requires input data to be in form graph which can be well presented using the adjacency matrix. To obtain an adjacency matrix from an image, a segmentation algorithm will be used followed by another algorithm making the adjacency matrix from the segmented images.

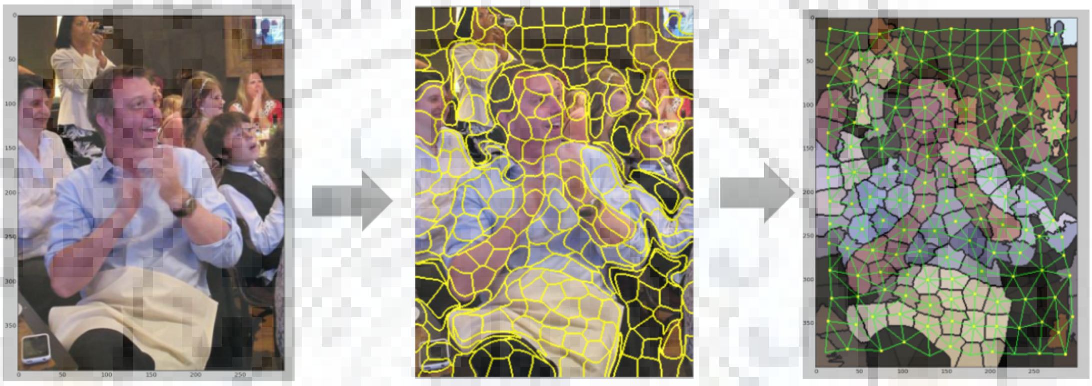


Figure 1.1: Preprocessing of an image to a graph with vertices as superpixels

In Figure 1.1, it is presenting the preprocessing required for the proposed methods. The image is first segmented using SLIC algorithm where we get the image with 100 segments consisting of super pixels and then the graph is constructed with the obtained super pixels over the image using RAG(Region Adjacency Matrix) which is suitable and as required by our proposed model.

In this method, using graph signal processing in irregular domain for CNNs we will be analyzing two different graph pooling operators and will give the impact of interpolation for recognizing local filters in the graph spectral domain[39]. These calculations are proved to be consistent with the earlier methods, hence beneficial for the network. Results are evaluated on the Pascal VOC2012 and MNIST dataset on irregular grid.

Chapter 2

Related Work

2.1 Literature review

Most of the real-world important datasets appear in an arrangement like a graph or network [25] like WWW, molecule interaction and social network etc. Yet till now, only a slight interest has been shown to the generalization of Convolution neural network model to such type of datasets.

In recent times, few researchers highlighted the issue of generalizing convolution networks to work on an arbitrary and irregular structured graphs and few of them are coming with better results in the spectral domain of graphs[5]. Some researchers are utilizing graph convolutions popular from the spectral theory of graphs to give description of filters that are applied in a neural network model [36, 38].

Now-a-days, most of the work is being done in a direction to fulfill the rift between slow and fast heuristics spectral methods. Deferred et al.(NIPS 2016) used the Chebyshev polynomials with parameters that are being learnt in a model like neural network to approximate filters in the graph spectral domain and attained promising output on MNIST dataset(regular domain), close enough to those of a simple two-dimensional convolutional neural network model. It is important to mention that the present neural network models faces some sort of issues when applied to regular graphs like chains or grid etc. Deffered et al. (NIPS 2016) gave a solution but a local spectral reduces to rotational symmetric filters and can never perform operations the same way a simple 2D CNNs on regular graphs. Also in the similar way, the Weisfeiler-Lehman algorithm also contradict when applied to regular domain .

Therefore, while coming for a solution for irregular domain it is necessary that it should be applicable to regular domain problem in the same way as simple 2D CNNs on regular grid [16]. Hence, a universal solution should be brought up which applicable for both regular and irregular spatial domain. In this work [30] they proposed a model for leaning CNNs for any random graphs. As in the image based classification problem the convolutional networks were relying on the regions that are connected locally in the input images but this work focus on the networks that are based on the regions that are connected locally from the graphs. Basically this works resides on the two reciprocal methods, firstly choosing a sequence of the vertices or nodes that cover up the larger part of the input graph and secondly, producing local normalized representation of the neighborhood for each node in the obtained sequence. It proves to be competitive with the classical graph signals but does not consider the stationarity property of the input data to construct the convolutional filters.

The authors of [5] generalizes the convolutional neural network to signals defined on the high dimensional data domain without using the translational operations. They brought forward two main frameworks where one is based on the clustering of domains in an hierarchical way and another uses graph laplacian spectrum. They built an efficient deep network which learn convolution with a count of parameters without depending on the input size.

Further, [16] improves over the [5] by including a graph estimation method as extension to the spectral network and tested the improvement on a large classification problems, finding that it matches and enhancing over the dropout Networks with the use of very few parameters for the estimation. But this work faces high cost of computation in matrices, both in forward and backward propagation.

The Graph based CNNs[12] introduces the graph convolution and the pooling operators over graph like the two over the regular domain. They calculated the gradients over the filters and input data, resulting in the generalization of the Convolutional Neural Network over the non-Euclidean irregular domain of input data. Their proposed graph CNNs when experimented over MNIST dataset, produced almost same result as obtained from the standard CNNs in the regular domain of input data. But this model demands expensive forward and inverse graph Fourier transform and requires some optimization over the model, encouraging for the future scope.

Recently, in [10] proposed the base of an efficient generalization of CNNs mathematically and computationally using the tools and techniques from the graph signal processing. They modeled an efficient architecture that extract feature by exploiting both the local and stationarity properties of the input data through graph convolutional layer. This work is more efficient as it avoids the specific us of Fourier basis over graph and hence gives better accuracy.

2.2 Graph Signal Processing

The arising field of graph signal processing draws the focus at filling up the gap residing in between the spectral theory of graph and signal processing[2, 7, 26] by combining the graph theory and the harmonic analysis. The main focus is to extend the basic analysis operations applied on the signals from the low dimensional regular grids to high dimensional irregular grids represented through graphs.

The classical operations, like convolution, filtering, down sampling and translation which are designed according to grid like structures are not directly applicable for the graphs and it requires modification in the mathematical definitions of these operation keeping the original concept in mind. In this framework, many researchers of[13, 15, 29] regained the knowledge about the formulation of wavelet operation on graphs and the methods to perform the multi-scale pyramid transform on graph the were depicted in [35]. After that the work of [31] reformed the uncertainty principle on graph and represented that some perceptive notion may be depreciated, but derives modified and effective localization principle.

2.3 CNNs on the Non-Euclidean Domains

Earlier, this concept arises from the Graph Neural Network(GNN) model[33] which further made simpler as Gated Graph Sequence Neural Network in [23]. This model was produced to enclose each vertex in an Euclidean space with a Recurrent Neural Network and further it utilizes these mappings as features used for the classification of the graph. It uses simple diffusion with a recursive relation to set the *transition function* f rather than using neural network. Hence their *state* vector will be s = f(x) = Wx and the output function g_θ will be set as $\hat{x}= g_\theta(s,x)=\theta$ $Lx+x$. With the help of K layer of GNN it can obtain the Chebyshev polynomial of degree K, further followed by a layer which is non-linear and the applying graph pooling. Hence this model can be elucidated as multi-layers of diffusions and local operations.

There after the author of [9, 14] came up with an idea of designing receptive fields locally to decrease the count of the parameters learned. The concept behind this is that on the basis of similarity measures grouping the features together making possible to choose only small number of connections between two consecutive layers. So far this model made a significant remark in reducing the count of parameters by utilizing the locality assumption but it did nothing to make use of any stationarity property, in short no weight sharing method. Then the authors of [5] utilized this concept for the spatial construction of graph based CNNs. Taking the benefit of stationarity property they utilized weighted graph to describe the local neighborhood and for pooling they calculated the multi-scale clustering of the graph. But the involvement of the weight sharing graph in the spatial formulation proved to be very challenging because it demands the ordering and selection of the neighborhood when the ordering (spatial or temporal) is missing as per the problem specifies to be.

The work of [27] introduces the generalization of CNNs to a smooth low dimensional and irregular domain, called the 3D-meshes. The deep learning architecture designed in this work defines the convolution operation on the mess patches using geodesic polar coordinates and produced the state-of-art outputs for the three dimensional shape classification. The first proposal of spectral construction of a graph CNN in [5], describes a filter or kernel. This concept of[16] although did a remarkable work but unable to scale up because it is based on the graph Fourier basis U which demands huge multiplication. Not only the cost of evaluating the matrix which needs eigenvalue decomposition on the Laplacian matrix of graph, the major contribution to the cost is that it need to multiply the input data with the same matrix twice both for the forward and the backward pass. For the smoothness in this domain their work depends on the spline parameters to exploit the locality of the data, but their model lacks control on the local support of their filter, which is required to learn the localized filters.

Chapter 3

Problem Formulation

3.1 *Problem Definition*

The well-known architecture of a convolutional neural network include an input layer, a group of convolutions and pooling layers and then a fully connected layer of neural network and an output layer for prediction of classes. But this architecture faces a major problem in the spatial domain with irregular structure of data. So, if the graph signal processing tools used along with the deep learning architecture then the solution for classification and regression in the irregular domain can be found. Hence to overcome this problem through graph signal processing methods it uses the multiplication in the spectral domain and extract features in the spatial domain by applying convolution for the irregular spaced domain, still features related spatially.

The problem definition states that we will be building a deep neural network which can be applied to data residing in an irregular or non-euclidean domain. The input will be in form of images and our model will recognize that image correctly as to which class it belongs. So firstly, we are working on the preprocessing of data to make it suitable for our network and feeding this preprocessed form of image to the convolutional neural network followed by the classifier which performs the image recognition task.

Figure 3.1 briefly describes the components of the Convolution neural network for graphs which resembles with the well-known standard CNNs and the convolution

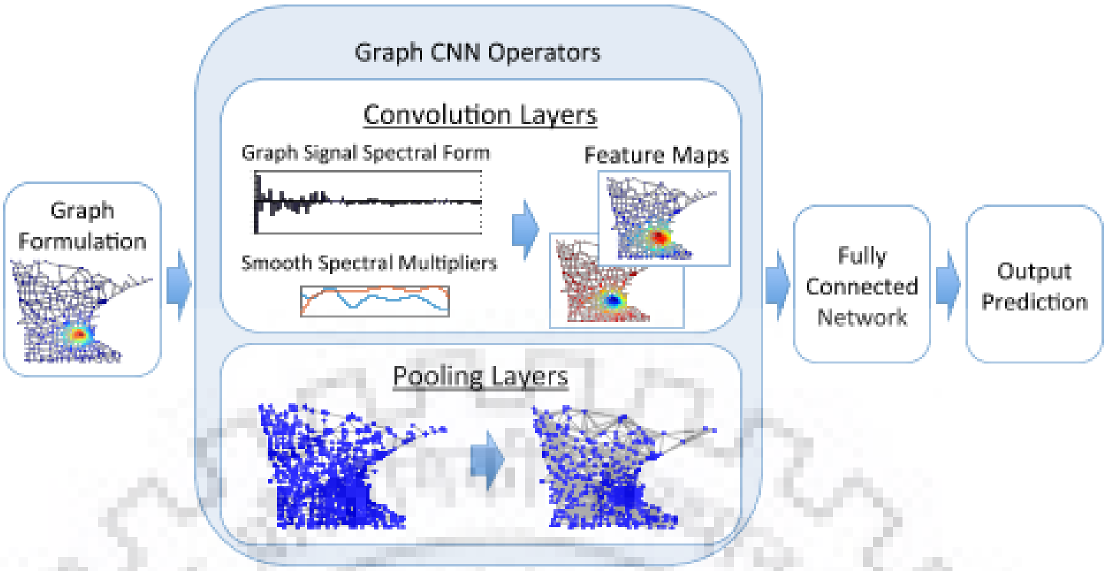


FIGURE 3.1: Graph based Convolutional Neural Network components

are learned based on the randomly initialized multipliers in the spectral domain and pooling layer based on graph coarsening[12] . The forward pass in the above network will be training of input data to get outputs and the back-propagation of the loss generated in the feed-forward pass during training updates the weights which were initialized randomly earlier.

The primary issue that comes to light while generalizing Convolutional Neural Networks to graphs, an irregular structured data is how to define the localized filters over graph. Therefore, the proposed neural model will handle all this issues and produced better result for both the regular and irregular structured data.

A graph G, undirected and connected, for a signal $f{:}V \rightarrow \mathbb{R}$ forms a carrier where the signal resides on the vertices of graph and f_j depicts the function value of signal $f \in \mathbb{R}^V$ at the j^{th} node or vertex in the graph. Graph signal processing allows various general operations like wavelet, translations, Fourier Transform and convolution on the signal f through G as the graph constitute the background knowledge regarding the spatial relationship between the nodes or vertices very well. It is possible to represent the problem domain in form of graph and hence possible to apply signal processing operations on the data in form of graph signal [15]. The important operator used for the analysis in the spectral domain is the graph laplacian.

Thus, this problem definition couples together deep learning architecture with the graph signal processing and helps in learning within the domains that are irregular, yet spatially related features and over which it is difficult for the standard Convolutional Neural Network to convolve using the regular kernel. So this problem statements highlight the fact that for making our CNN to convolve over irregular domain, the preprocessing over the input is required, and considering the preprocessed input data the convolution layer, pooling layer is defined.

Chapter 4

Proposed Approach

4.1 *Methods*

Processing on graph signal as shown in Figure 1.1 along with deep learning can make it possible to learn in the spatial domain with irregularly structured data over which the general well-known CNNs is unable to convolve [39]. The proposed approach will help in a direction where using deep learning in pattern or image recognition domain its possible to convolve across domain with irregularity but features are spatially related[15, 39]. The generalization of Convolutional Neural Network to graph structured data demands three basic steps: (i) the convolutional filters localized and designed over graph signals, (ii) graph coarsening method that can associate together vertices on the basis of some similarity and (iii) a pooling method for graph which attempt to reduce the input feature map's resolution and generalizes the identified features well.

4.1.1 Graph Convolution

Majority of real-datasets are unstructured and can be structured with graphs because graph can easily model the heterogeneous relationships between the vertices and can encode even the complex geometric structured data domains.

To define the convolution filters over graph there are two approaches, that are spatial and spectral approach. Through spatial approach the localization property can be satisfied as it provide the filter localization through the finite size of the filters. Though graph convolution in the spatial domain is quite reasonable but it has to face the challenges to match the neighborhood locally, highlighted in [5]. Along-with this issue, the spatial domain lacks the exclusive definition of translation over graph under mathematical terms. While the spectral as the second approach brings well defined localized convolutional filter in the spectral domain which are linear operators diagonalizing in the Fourier basis. But, because of multiplication in the Fourier basis , the convolution and the translation in this domain are very costly. These issue with both the approaches can be resolved using filter parametrization.

Graph Fourier Transform

In signal processing, where the signal characterized over undirected and connected graph G={V,E,W}, where V is the set of vertices such that $|V| = N$, E is the set constituting all the edges and $W \in \mathbb{R}^{N*N}$ is an adjacency matrix having weight entries with respect to vertices u_i and u_j if there is non-negative, undirected and non-self-looping edge between them. In spectral graph theory, the graph Laplacian L matrix being an important operator in the analysis can be illustrated as $L = D - W \in \mathbb{R}^{N*N}$, where $D \in \mathbb{R}^{N*N}$ the diagonal matrix where the entry $d_{j,j} = \sum_{i=1}^{N} W_{j,i}$ is the degree of vertex v_j. Laplacian matrix fundamentally tells how smooth the function over graph is, that is the value of function remains almost constant for the connected vertex.

The data signal $f \in \mathbb{R}^N$ lies on the graph G, where the value of j^{th} component of vector$f$ is f_j which resembles with the amplitude of the signal at the v_j . The normalized form of Laplacian matrix $L = I_n - D^{-1/2}WD^{1/2}$ and here I_n is an Identity matrix. Because the Laplacian matrix is a real symmetric ans positive semidefinite matrix, it's set of orthonormal eigenvector are defined as the Fourier Modes and the co-related ordered positive eigenvalues, determined as frequencies of the graph signal. To generalize the convolutions to random and more general graph it is necessary to find the respective ”Fourier” basis and in the spectral graph theory this correspondence is given by the Laplacian Matrix. So, Laplacian L is actually diagonalized through the Fourier basis and in the Euclidean space, the graph Fourier transform will authorize the construction of basic operation such as filtering.

Convolution Layer

The CNN architecture shown in the Figure 4.1 has three main operations that are graph pooling, coarsening and convolution. The Convolution layer extract features present in the input image. In the graph CNN we can perform convolution like in standard CNN by applying convolution theorem in the regular domain as a multiplication in the frequency domain.

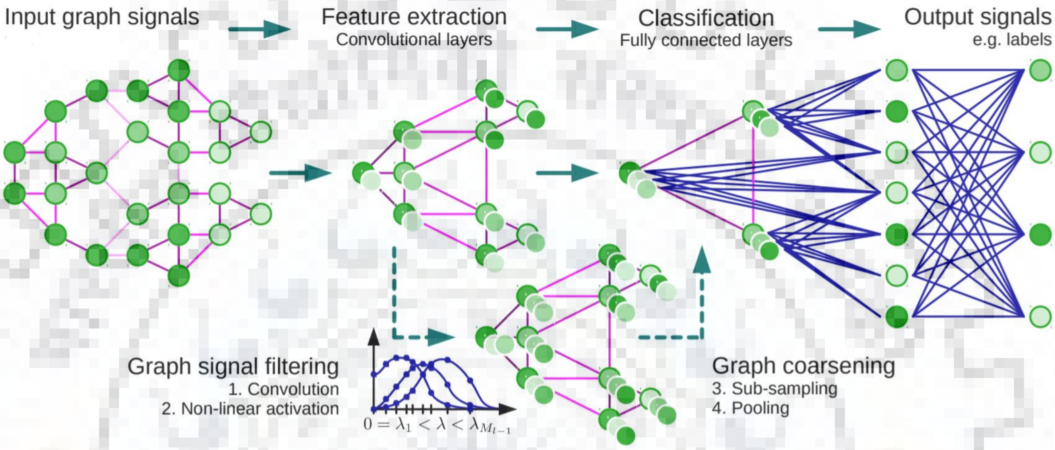


FIGURE 4.1: Graph based Convolutional Neural Network Architecture

Now, to map the signal on graph to the frequency domain the matrix L is needed to be broken up into eigenvectors matrix $X = \{x_{j=1.....N}\}$,where the x_j is the orthonormal vector with eigenvalues $\lambda_{j=1...N}$. For a given data signal the Graph Fourier Transform[4] is as

$$\tilde{f_j} = \sum_{i=1}^{N} \lambda_i f_j^T x_j \text{ and using matrix X, } \tilde{f_j} = X^T f$$

$$\text{and its inverse is } f_j = \sum_{i=1}^{N} \lambda_i \tilde{f_j} x_j \text{ is same as } f = X\tilde{f}$$

In the forward pass, the multiplication of Laplacian operator in the Fourier domain space will be the convolution operator. Using Graph spectral domain's [8] concept, the convolved signal as output will be spectral multiplication as $z = X\tilde{f}k$ where the $\tilde{f}$ and k are the spectral graph signal and multiplier respectively. If the input is given through multiple channels then there will be multiple output feature maps:

$$z_{s,o} = X \sum_{j=1}^{I} X^T f_{s,j} \odot k_{j,o}$$

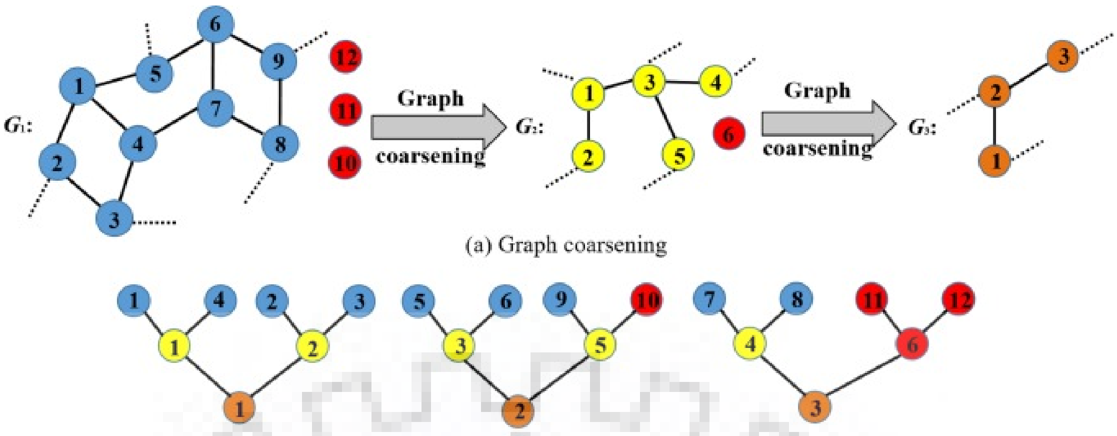
Here, I is the total count of input channels for signal f and o is the corresponding output feature for input j signal and s is the particular batch sample. For the graph CNNs, the vertex localization in the spatial domain is provided by smoothness within the spectral region. Hence to extract the local features in the regular domain the spectral multipliers involved in the convolution are recognized by following sampled set of filter weights $\hat{k}_{j,o}, k \in \mathbb{R}^{<N}$ which then goes under interpolation till it become the complete filter through smoothing kernel Φ like cubic splines: $k_{j,o} = \Phi\hat{k}_{j,o}$ [12, 28]. This operation have both advantage and disadvantage, disadvantage as it adds couple of extra operations in the interpolation to full filter for multiplication and advantage as reduced the count of tracking weights resulting in less tuning parameter and therefore, learns sharper feature maps.

4.1.2 Graph Coarsening

In pooling similar vertices need to be clustered together, which in turn requires some similarity between the neighboring vertices. If the same is applied for the multiple layers, it resembles the concept of multi-scale clustering of graphs maintaining the local geometrical structure. Being an NP hard problem Graph clustering requires approximation. Many clustering techniques are available, but we are more concerned over multi-level clustering techniques where a coarser graph is produced at each level corresponding to the data domain, if viewed at different resolution.

For better understanding of graph coarsening consider a graph $G = \{V, W\}$ and it is coarsened to a graph $\hat{G} = \{\hat{V}, \hat{W}\}$ where the $|\hat{V}| < |V|$ and also those edges connecting the removed vertices are also removed. Graph coarsening [6] results in a sparse graph by constructing $\hat{V}$ by either selecting only a subset of V or generating new set of $\hat{V}$ by aggregating related vertices in V. A better solution to select $\hat{V}$ is to select the maximum eigenvalue λ_N and dividing the rest of the vertices in V according to the sign of eigenvector X_N [32].

Using any clustering methods that can decrease the graph size by half at each level provides better control on the size of coarsening and pooling. We are using Graclus multilevel clustering algorithm for coarsening of graph, which has been found to be highly effective for clustering wide range of graphs. Figure 4.2 explains the working of graph coarsening and pooling.



(a) Graph coarsening

(b) Pooling

FIGURE 4.2: Graph Coarsening and Pooling

Graclus[11], formed on Metis [18], consequently computes coarser version for a graph using the greedy approach and has the capability to optimize various common objectives of spectral clustering, out of which normalized cut [34] is used in this proposed work. Greedy approach mentioned above selects an unvisited node i and matches it with other unvisited neighboring node j such that normalized cut $W_{ij}(1/d_i + 1/d_j)$ is maximized locally. It marks the two matching vertices as visited and their sum of weights will be the coarsened weight. The same procedure repeats until all vertices have been visited. It reduces the number of nodes to almost half proceeding from one level to another, making this algorithm highly efficient and computationally fast.

4.1.3 Graph Pooling

After Convolution layer the, the other component of the convolution neural network is the pooling layer which is responsible for decreasing the resolution of the feature maps in order to generalize the features extracted after the convolution in the spatial domain of input data and to handle the space complexity while using the various filters [32]. As we know that the convolution of graph involves the multiplication of $\mathbb{R}^N$ filters with almost same number of signals and hence there is no reduction in the size of graph from input signal till output features.

Therefore, the pooling layer comes into picture in the graph convolutional neural network to handle the space complexity by applying an algorithm which can cache single instance of the graph and its eigenvector matrix X having size $O(N^2)$. In graph CNNs [6], to pool the features together the coarsening of graph must be applied and then map the input features in the original domain of input data to the new graph with reduced size.

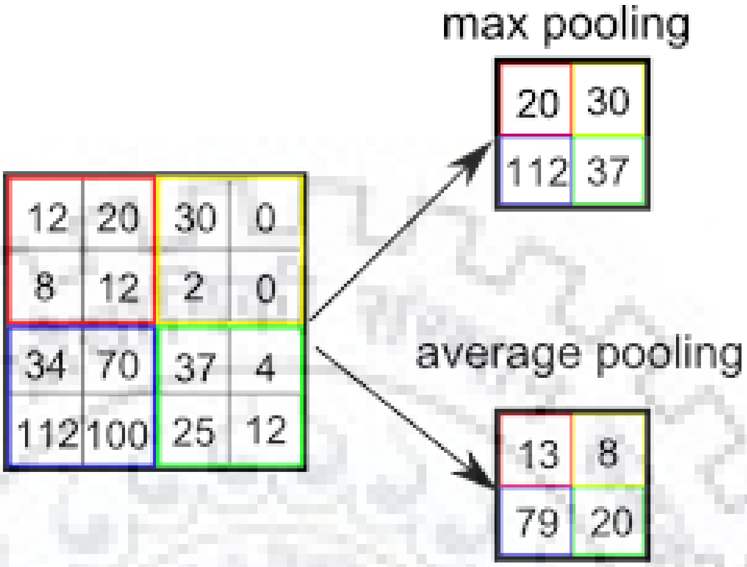


Figure 4.3: Pooling method(max pooling and average pooling)

In Figure 4.3 and 4.2 shows the graph pool methods used in our architecture. In the max pooling method, the maximum value in the sub-sampled grid is taken as the pooled value while in average pooling, the pooled value will be the average value of the respective sub-sampled grid in the feature maps.

In this proposed model, the pooling operation is applied many times, thus need to be effective. After the graph is coarsened, the produced vertices from the input graph are not systematically arranged. If we apply the pooling layer to the coarsened graph directly, it will require an extra storage to store the matched vertices or the coarsened vertices. This implementation will not be efficient in terms of both memory and computation, as it is difficult to run it parallelly.

However by arranging the vertices will make the pooling operation as effective as linear pooling. It comprises two steps to be followed sequentially. Firstly, a balanced binary tree is created, and then the vertices are rearranged in the second step. When the graph is coarsened, every vertex in the resulting graph will either have two children, if there is a finer match, else it will have one child. Dummy nodes, or disconnected node are added to pair up with the singleton node, providing each node two children, resulting in the balanced binary tree.

In this resulted binary tree, regular nodes will be having either both children as regular nodes, or a pair of dummy and singleton as children. But the dummy node will always have dummy nodes as children.

The input signal lying over the graph will initially have neutral value for these dummy nodes, which remains unaffected during filtering, while using a ReLU activation layer with max pooling layer as they are disconnected. Although, addition of dummy nodes result in increased dimensionality and thus increases the computational cost. But, in practical scenario, the count of singleton nodes is very less using Graclus coarsening technique.

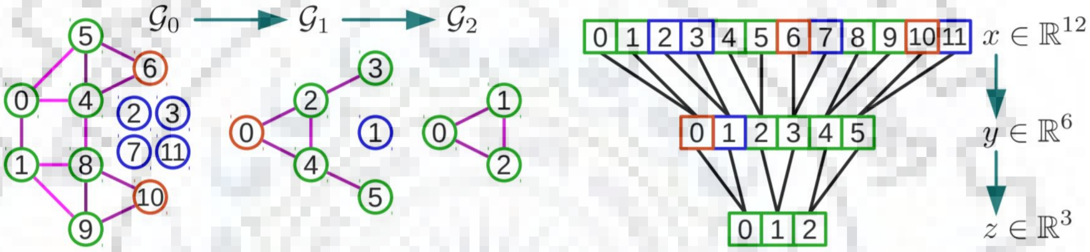


FIGURE 4.4: Graph Coarsening Method and Pooling

A random arrangement of nodes done at coarsest level, and forwarding the same towards the finest levels, results in a regular ordering at the finest level. Here regular ordering refers to the hierarchical merging of neighboring nodes at the finest level. Pooling operation over such graphs is similar to that of pooling a 1D signal. The whole process is illustrated in Figure 4.4. Regular ordering increases the efficiency of pooling operation and enables parallelization making it work fast as memory accesses are spatial in nature, since fetching of matched nodes are not required.

Figure 4.4 shows an example of Graph Coarsening and Pooling. Here, let the graph G_0 is given as input which is the finest graph and $x \in R^8$ be a signal residing on the graph G_0. The graph G_0 having 8 vertices $|V_0| = n_0 = 8$ which are arranged in an arbitrary manner. And apply a pooling operation of size 4 or two pooling, each of size 2 on x. The coarsening algorithm used, Graclus provides the coarsest graph G_1 and G_2 of size $n_1 = |V_1| = 5$ and $n_2 = |V_2| = 3$ respectively. Thus few dummy nodes are added to both graph G_1 and G_2 to couple with the singleton nodes, so that every node has both the child nodes. This set the size of graph G_0, G_1 and G_2 to $n_0 = 12$, $n_1 = 6$ and $n_2 = 3$.

Then the vertices in V_2 are arranged arbitrarily and vertices in V_1 and V_0 are arranged in the same manner in consequent way. The order of vertices in V_0 allows a regular 1D pooling on $x \in R^{12}$ at this particular moment, so $z = [max(x_0, x_1), max(x_4, x_5, x_6), max(x_8, x_9, x_{10})] \in R^3$.The value of x_2, x_3,x_7, x_{11} are assigned a neutral value as these are signal component that are added as the dummy nodes.

4.1.4 Back-propagation on graphs

Initially the weights in the graph are randomly initialized, therefore when the errors are propagated in the network from the output to input layer these randomly initialized weights and bias are updated to achieve target function along with gradient descent. Back-propagation of loss or errors play a central role in the deep learning[16].

For back-propagation of errors the derivatives with respect to weights and input is needed to get output, in graph CNNs the gradients are framed keeping in mind the graph signal f and the multiplier k [16]. For particular channel , the input feature map gradient $f_{s,j}$ is the convolution of output gradient and the spectral multiplier given batch of S graph signals through:

$$\Delta f_{s,j} = X \sum_{o=1}^{O} X^T \Delta z_{s,o} \odot k_{j,o}$$

For full set of multipliers which are interpolated, the gradient is convolution of output gradient and input signal for given batch sample s through:

$$\Delta k_{j,o} = \sum_{s=1}^{N} X^T \Delta z_{s,o} \odot X^T f_{s,j}$$

All the filters are nothing but the multipliers, so it is not required to propagate back via graph Fourier transform. The smooth multiplier Δk can be back-propagated by multiplying with the inversed smooth kernel through $\Delta \tilde{k}_{j,o} = \Phi^T \Delta k_{j,o}$.

Chapter 5

Experimentation and Results

5.1 *Data-set*

The graph CNNs [16] are basically postulated to irregular structure domain issues more; like sensor networks, social networks, human activity graphs, molecule interaction graphs, etc., but it is non-trivial to characterize kernel filters used for convolution in this domain. Hence, the proposed graph CNNs architecture will be evaluated on the standard 2D matrix of both regular and irregular input data domain.

The 2D matrix is a graph representation which carries information of relationships between pixels of an image. Graph G on an image, each vertex is denoted by a pixel and the corresponding intensity of each pixel at each vertex for a graph signal f. The weights between the edges in the adjacency matrix is calculated as the euclidean distance between the vertices . Our proposed model has been evaluated on two dataset- MNIST and PASCAL VOC2012.

MNIST

The MNIST is the collection of handwritten digits, which is a subset of bigger dataset accessible form NIST. The digits present in this dataset are normalized with respect to the size and centered in an image of fixed size. To evaluate this architecture [12] on 2D matrix we will be using MNIST dataset containing 60,000 training and 10,000 testing example of hand-written digits in 28×28 matrix of gray-scale pixels. The edge weights are the euclidean distance between the vertices in the graph with $f \in \mathbb{R}^{784}$.

As our main focus is to evaluate performance of the graph based CNNs on the irregular domain, so by sub-sampling the above grid by removing random vertices from graph and their corresponding edges we can obtain an irregular spatial domain such that now $f \in \mathbb{R}^{700}$. The irregular spatial graph can be fed to the Graph CNN to get convolved output feature maps.

PASCAL VOC2012

Pascal VOC2012 dataset containing 20 classes and the training and validation data has approximately 11,500 images having around 27,400 ROI annotated objects and almost 6,900 segmentations. The new development made over the previous version pascal VOC datasets is that the size of segmentation dataset has increased considerably. For classification problems, detection and person layout the dataset is same as VOC2011.

For training and validation purpose, the 50% dataset has been used while rest used for the testing purpose. Among the train set, validation set and test set the dispersion of the images and the object by the class are made uniform, for better efficiency of the model.

5.2 Implementation

The network implemented using Tensor Flow with GPU enabled operators is trained for 1,500 iteration on MNIST dataset and 50,000 iterations on the PASCAL VOC2012 dataset to test the samples in test data at each iteration to note the efficiency of the model. The upcoming section will give better insight of each stage of the prosed approach.

Segmentation

As our dataset is collection of images, and our graph CNN model require the input in the form of graph, well represented with an adjacency matrix. To acquire this adjacency matrix from the image some preprocessing over the image is required. Firstly, we cannot generate the graph from the image directly by taking all the pixel as vertices, because it will be inefficient both in terms of memory and computation. So, an efficient approach of segmentation is needed.

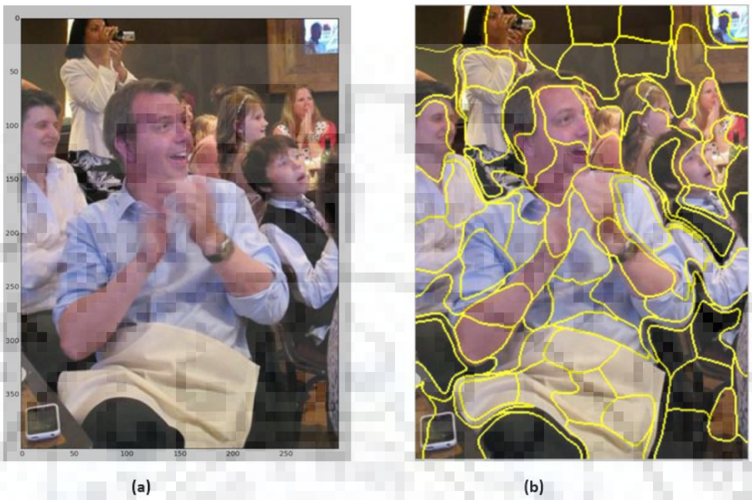


Figure 5.1: SLIC based superpixel segmentation

Simple Linear Iterative Clustering (SLIC) [1] is the most popular algorithm used to segment an image in super pixels, as it is computationally efficient. Superpixels, are better explained as the group of pixels which share some common properties. In short, SLIC clusters pixels in a 5-D space to effectively produce the compact, and almost uniform superpixels. Another advantage associated with this technique is that it encapsulates both the pixel as well as grey-scale image. In figure 5.1 on the basis of similarity in color and proximity among the pixels they are clustered in a 5-D space $[labxy]$ where $[lab]$ is a pixel color vector in CIELAB color space. Here (a) in figure is the input image over SLIC is used to get 100 segments or superpixels and (b) is the resultant image with 100 segments.

SLIC offers many advantage over the other proposed approaches, low complexity $O(N)$ where N, is the number of pixels present in the image, a precise control over number of superpixels and compactness among the superpixels required as per need of the user. After retrieving the superpixels, an adjacency matrix is maintained to feed the network.

Graph CNN Model

Figure 5.2, shows the architecture of our model which comprises of the input adjacency matrix fed to the convolution layer 1, i.e gcnn_1 which produce feature maps through 64 output channels and which further fed to second convolutional layer gcnn_2 producing through same number of output channels as in the first layer. Then a Max pool layer,i.e maxpool_1 is applied with pool size=4 then the pooled features are fed to the next two convolution layer gcnn_3 and gcnn_4 consecutively followed by a max pooling layer and then an average pooling layer. The pooled feature maps are then fed to the fully connected layer fc_1. The gradient and accuracy is calculated side by side.

Adam [19] is an algorithm which is utilized for the optimization of first order gradients- based objective function. The algorithm is easy to implement as it is computationally fast, requires small space and unaffected with the rescaling of gradients diagonally. This is very useful for those problems which are focusing on the large range of dataset.The entire model is implemented using tensorflow implementation.

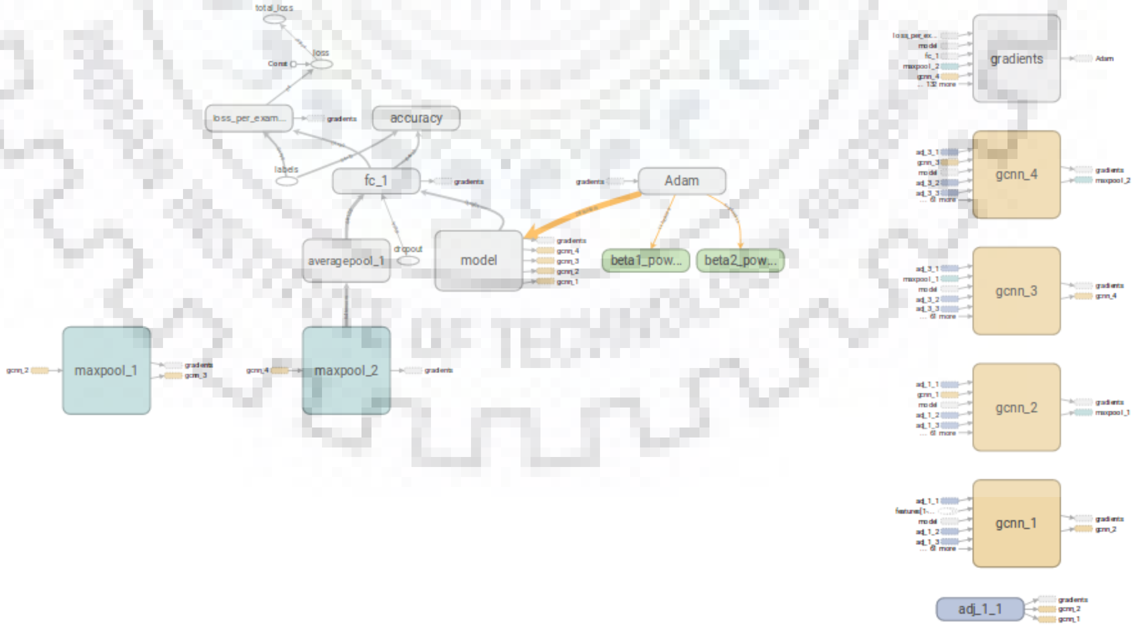
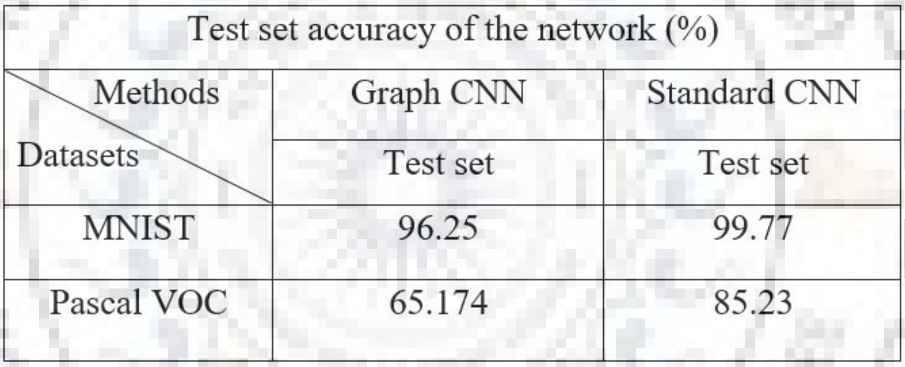


Figure 5.2: The graph CNN Architecture

The model is trained over both the dataset MNIST and PASCAL VOC2012 with a learning rate=0.001 and batch size=64. Softmax_cross_entropy_loss is used to calculate the error over the softmax layer used as the activation layer.

5.3 *Results*

The graph based CNN was evaluated using both the dataset on their irregular matrix .The table1 will show comparatively the performance of standard CNN and graph based CNN on both the datasets[12]. The efficiency of the model while performing the testing on the samples which were not observed earlier is very hopeful to work in this field as it provides better output features as compared to standard CNN if irregular domain is considered. The efficiency of the classification on both the dataset is detailed in the Table 5.3 and presented in the Figure 5.8 shows the comparison between standard CNN and graph CNN. The proposed graph CNN model achieves 96.25% accuracy on MNIST dataset and 65.174% on Pascal VOC dataset. Although the accuracy achieved is less as compared to the standard CNN architecture but it is acceptable as the proposed model will be applicable for both the domain that is regular and irregular domain while the standard CNN was applicable to only regular spatial domain.



Test set accuracy of the network (%)		
Methods Datasets	Graph CNN	Standard CNN
	Test set	Test set
MNIST	96.25	99.77
Pascal VOC	65.174	85.23

FIGURE 5.3: Result showing test set accuracy of the network

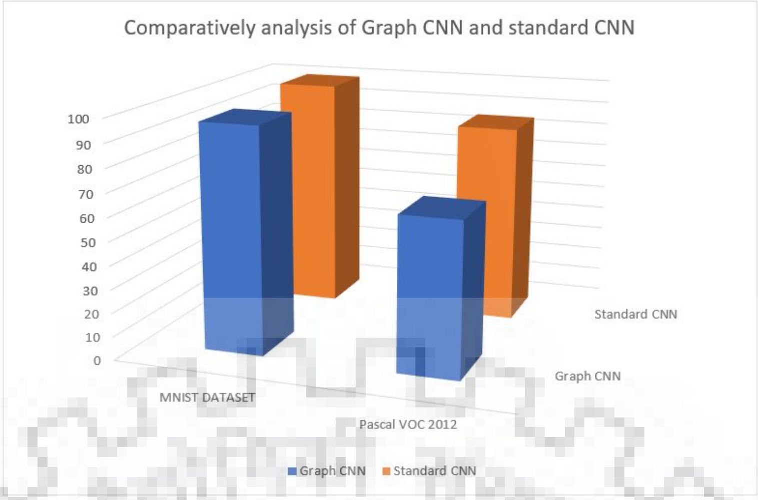


Figure 5.4: Bar-Graph presenting performance of the both Graph CNN and standard CNN

MNIST

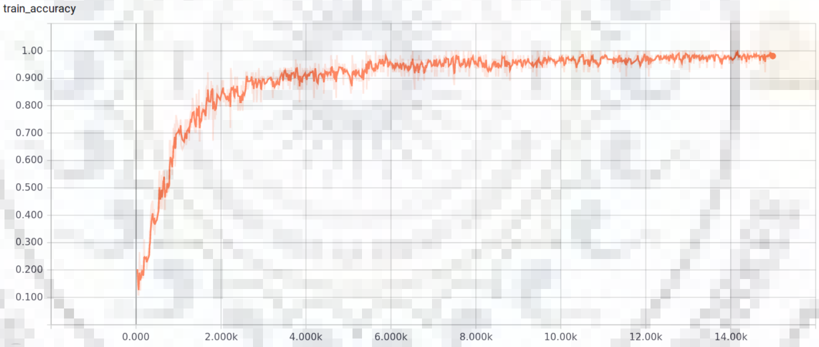


Figure 5.5: Training Accuracy curve over MNIST

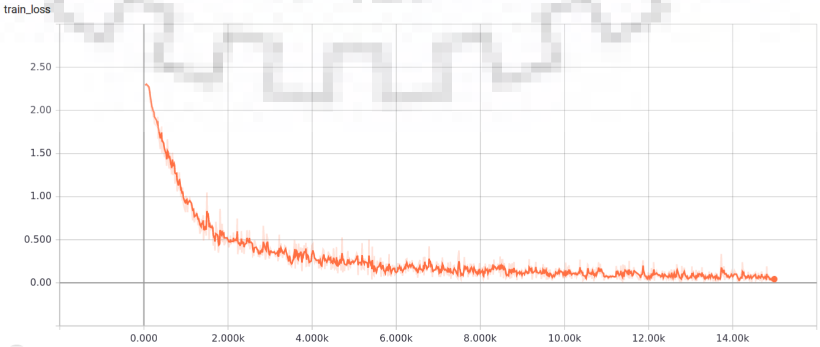


Figure 5.6: Training Loss curve over MNIST

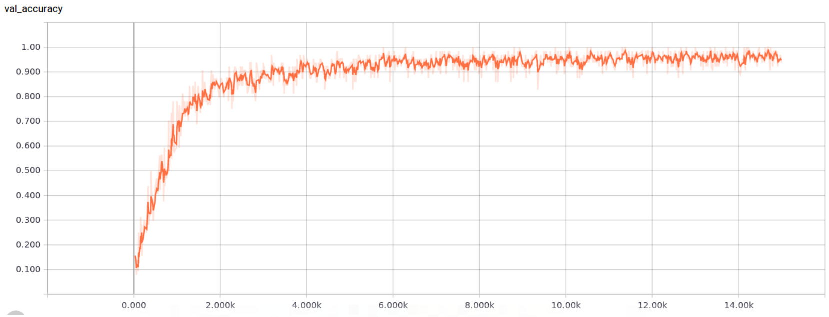


Figure 5.7: Validation Accuracy curve over MNIST

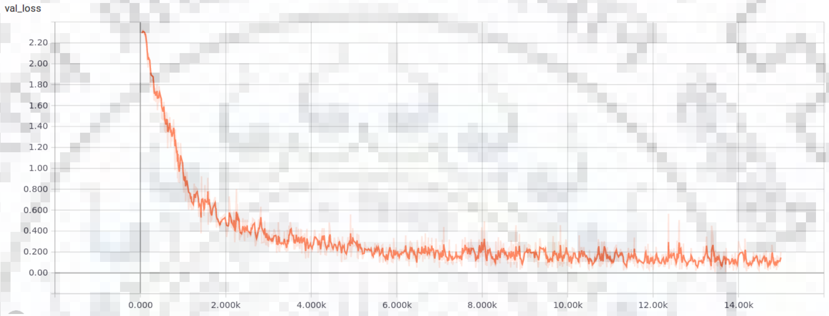


Figure 5.8: Validation Loss curve over MNIST

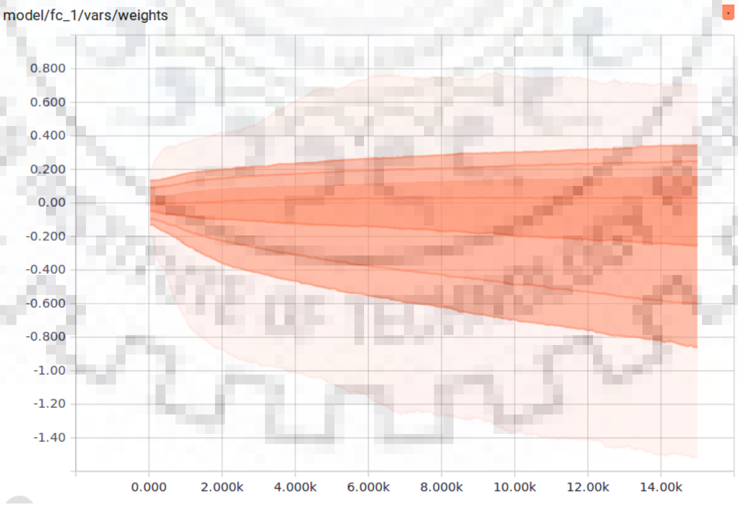


Figure 5.9: Weight of Fully connected layer

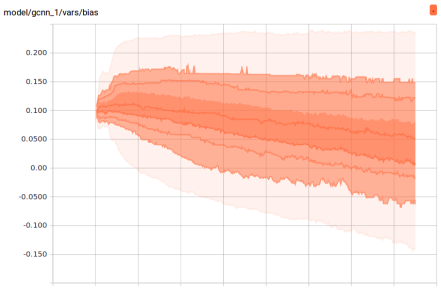


Figure 5.10: Bias of Convolution layer 1

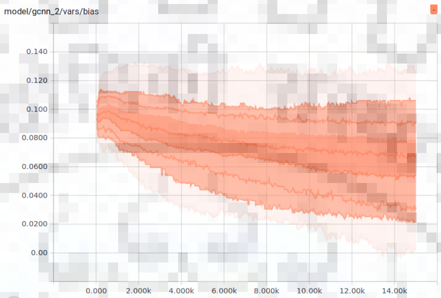


Figure 5.11: Bias of Convolution layer 2

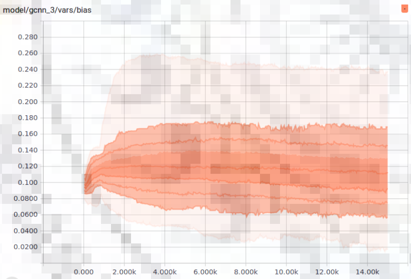


Figure 5.12: Bias of Convolution Layer 3

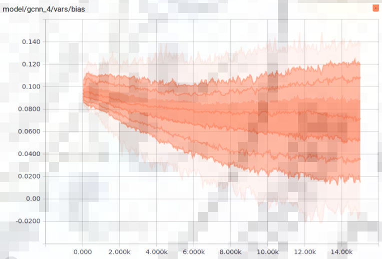


Figure 5.13: Bias of Convolution Layer 4

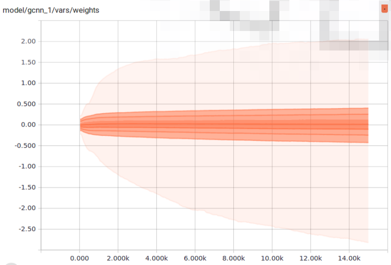


Figure 5.14: Weights of ConvLayer 1

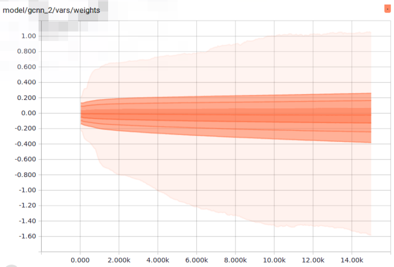


Figure 5.15: Weights of Conv Layer 2

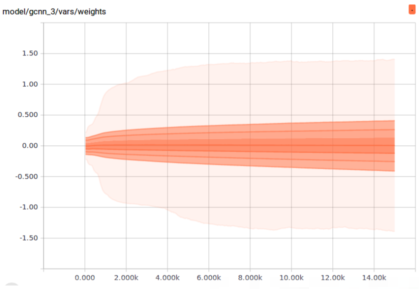


FIGURE 5.16: Weights of ConvLayer 3

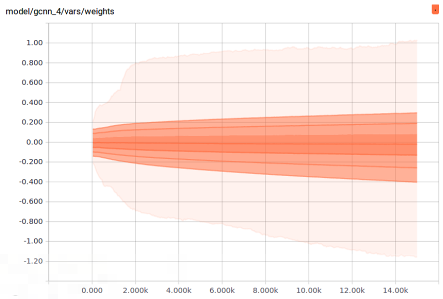


FIGURE 5.17: Weights of ConvLayer 4

Pascal VOC2012

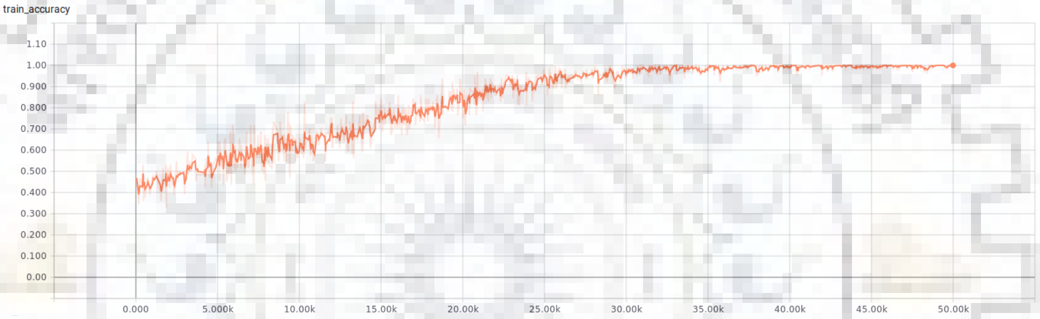


FIGURE 5.18: Train Accuracy Curve over Pascal VOC2012

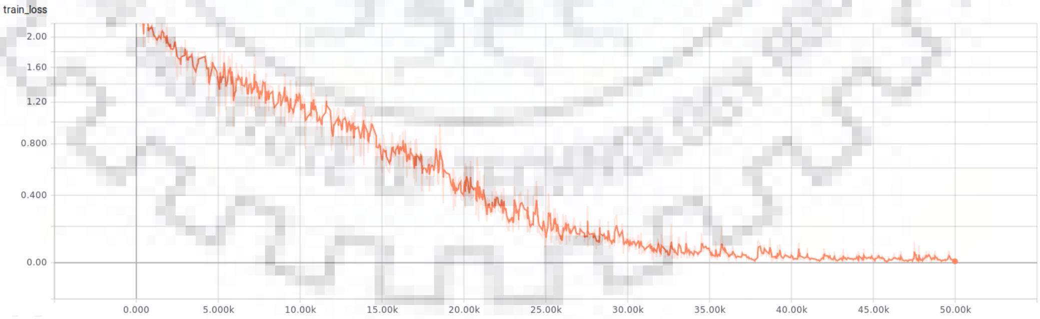


FIGURE 5.19: Training Loss Curve over Pascal VOC2012

Conclusion and Future Scope

The idea introduces a new approach of performing convolution on both regular and irregular graph by combining graph signal processing methods and neural networks based on the back-propagation. It meets the requirement of learning convolution on the graph Laplacian matrix and extract the features even after the non triviality of the kernel design for the irregular domain. Results are given for the performance of graph CNN on the irregular graph input domain and in the future, there is more scope of work on irregular graph based on real dataset from social networks and improvement in the performance over this domain. Graph pooling method is given to reduce the size of the graph so that it can be easy to handle the space complexity of the graph and to generalize the feature maps. As on the regular graph, standard CNN is giving better result than the proposed method but standard CNN is applicable to only regular graph and the proposed method will be applicable to both the domain. Also the graph CNN involves Fourier transform and inverse Fourier Transform which are very costly operations, so it carries a future scope to enhance the methods of pooling and graph construction to improve the performance of the graph CNN in regular domain with respect to standard CNN as well as make it efficient for irregular graphs. Also a further scope of the problem involves study the same graph CNN method for directed or relational graphs as till now the whole discussion was for undirected and non-negative graphs.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, June 2003.

[3] Y.-L. Boureau, J. Ponce, and Y. Lecun. A theoretical analysis of feature pooling in visual recognition. In *27TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING, HAIFA, ISRAEL*, 2010.

[4] R. Bracewell. The fourier transform and its applications. *American Journal of Physics*, 34(8):712–712, 1966.

[5] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *CoRR*, abs/1312.6203, 2013.

[6] C. Chevalier and I. Safro. *Comparison of Coarsening Schemes for Multilevel Graph Partitioning*, pages 191–205. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[7] F. R. K. Chung. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, 1997.

[8] D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, June 2012.

[9] A. Coates and A. Y. Ng. Selecting receptive fields in deep networks. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger,

editors, *Advances in Neural Information Processing Systems 24*, pages 2528–2536. Curran Associates, Inc., 2011.

[10] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. *ArXiv e-prints*, June 2016.

[11] I. S. Dhillon, Y. Guan, and B. J. Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(0), nov 2007.

[12] M. Edwards and X. Xie. Graph based convolutional neural network. *CoRR*, abs/1609.08965, 2016.

[13] M. Gavish, B. Nadler, and R. R. Coifman. Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning.

[14] K. Gregor and Y. LeCun. Emergence of Complex-Like Cells in a Temporal Product Network with Local Receptive Fields. *ArXiv e-prints*, June 2010.

[15] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129 – 150, 2011.

[16] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

[17] G. Hinton, L. Deng, D. Yu, G. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *Signal Processing Magazine*, 2012.

[18] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International*

Conference on Neural Information Processing Systems - Volume 1, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

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

[22] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5325–5334, June 2015.

[23] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated Graph Sequence Neural Networks. *ArXiv e-prints*, Nov. 2015.

[24] P. Liu, X. Wang, and Y. Gu. Graph signal coarsening: Dimensionality reduction in irregular domain. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 798–802, Dec 2014.

[25] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[26] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, Dec. 2007.

[27] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on Riemannian manifolds. *ArXiv e-prints*, Jan. 2015.

[28] R. Merris. A note on laplacian graph eigenvalues. *Linear Algebra and its Applications*, 285(1):33 – 35, 1998.

[29] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *ArXiv Mathematics e-prints*, Mar. 2005.

[30] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *International Conference on Machine Learning*, pages 2014–2023, 2016.

[31] B. Pasdeloup, R. Alami, V. Gripon, and M. Rabbat. Toward An Uncertainty Principle For Weighted Graphs. In *EUSIPCO 2015 : 23rd European Signal Processing Conference*, pages 1496–1500, 2015.

[32] I. Safro, P. Sanders, and C. Schulz. Advanced coarsening schemes for graph partitioning. *J. Exp. Algorithmics*, 19:2.2:1–2.2:24, Jan. 2015.

[33] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *Trans. Neur. Netw.*, 20(1):61–80, Jan. 2009.

[34] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000.

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

[36] 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, May 2013.

[37] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *Proceedings of the 11th European Conference on Computer Vision: Part VI*, ECCV'10, pages 140–153, Berlin, Heidelberg, 2010. Springer-Verlag.

[38] M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *CoRR*, abs/1301.3557, 2013.

[39] M.-L. Zhang and Z.-H. Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351, Oct 2006.

Image Recognition using Graph Based Convolutional Neural Network

ORIGINALITY REPORT

2%	0%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

Submitted to Imperial College of Science, Technology and Medicine Student Paper	<1%
Inderjeet Kaur, Rana Gill. "Chapter 31 Super-Pixel Based Segmentation of Urban Area and Its Detection Using Machine Learning Techniques", Springer Nature, 2018 Publication	<1%
Xin Yao. "A novel memetic algorithm with random multi-local-search: a case study of TSP", Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat No 04TH8753) CEC-04, 2004 Publication	<1%
Submitted to University of Queensland Student Paper	<1%
"Computer Vision – ECCV 2016 Workshops", Springer Nature, 2016 Publication	<1%

Carla E. Brodley. "Solving cluster ensemble problems by bipartite graph partitioning", Twenty-first international conference on Machine learning - ICML 04 ICML 04, 2004
Publication
<1%

Huang, Jui-Ting, Jinyu Li, and Yifan Gong. "An analysis of convolutional neural networks for speech recognition", 2015 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 2015.
Publication
<1%

Submitted to Higher Education Commission Pakistan
Student Paper
<1%

cse.seas.wustl.edu
Internet Source
<1%

Pengyu Wang, Yuan Gan, Panpan Shui, Fenggen Yu, Yan Zhang, Songle Chen, Zhengxing Sun. "3D shape segmentation via shape fully convolutional networks", Computers & Graphics, 2018
Publication
<1%

Goldberg, Richard, Stephen Smith, Jack Mottley, and K Ferrara. "Ultrasound", Medical Imaging Principles and Practices, 2012.
Publication
<1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		