Spatio-Temporal driven Attention Graph Neural Network with Block Adjacency matrix (STAG-NN-BA) for Remote Land-use Change Detection

Usman Nazir,[1] **W. Islam,** [1] **M. Taj** [1] **S. Khalid** [2]

[1] Lahore Uniersity of Management Sciences
[2] University of Oxford
{usman.nazir, 20030031, murtaza.taj}@lums.edu.pk
sara.khalid@ndorms.ox.ac.uk

Abstract

Despite the recent advances in deep neural networks, standard convolutional kernels limit the applications of these networks to the Euclidean domain only. Considering the geodesic nature of the measurement of the earth's surface, remote sensing is one such area that can benefit from non-Euclidean and spherical domains. For this purpose, we design a novel Graph Neural Network architecture for spatial and spatio-temporal classification using satellite imagery to acquire insights into socio-economic indicators. We propose a hybrid attention method to learn the relative importance of irregular neighbors in remote sensing data. Instead of classifying each pixel, we propose a method based on Simple Linear Iterative Clustering (SLIC) image segmentation and Graph Attention Network. The superpixels obtained from SLIC become the nodes of our Graph Convolution Network (GCN). We then construct a region adjacency graph (RAG) where each superpixel is connected to every other adjacent superpixel in the image, enabling information to propagate globally. Finally, we propose a Spatially driven Attention Graph Neural Network (SAG-NN) to classify each RAG. We also propose an extension to our SAG-NN for spatio-temporal data. Unlike regular grids of pixels in images, superpixels are irregular in nature and cannot be used to create spatio-temporal graphs. We introduce temporal bias by combining unconnected RAGs from each image into one supergraph. This is achieved by introducing block adjacency matrices resulting in novel Spatio-Temporal driven Attention Graph Neural Network with Block Adjacency matrix (STAG-NN-BA). We evaluate our proposed methods on two remote sensing datasets namely Asia14 and C2D2. In comparison with both non-graph and graph-based approaches our SAG-NN and STAG-NN-BA achieved superior accuracy on all the datasets while incurring less computation cost. The code and dataset will be made public via our GitHub repository.

1 Climate Impact Statement

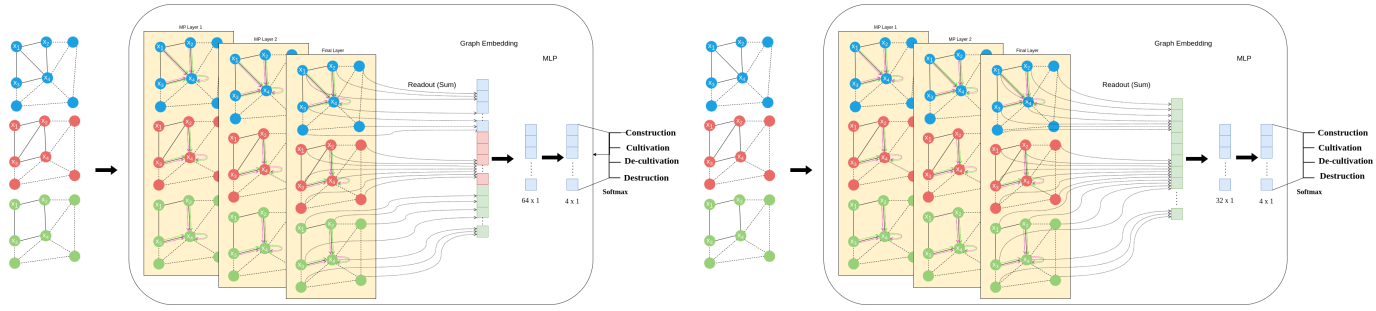
The proposed innovative approaches: SAG-NN and STAG-NN, aimed at harnessing the potential of spatial and spatio-temporal data, is not only limited to classification tasks but extends its utility to detect critical transitions. Specifically, transitions between classes such as construction, destruction, cultivation, and decultivation - representing fundamental human activities with historical significance - can be effectively identified (see Fig. 10 in Appendix). Thereby the proposed approach will contribute to enhanced understanding and actionable insights in addressing climate change and its complex ramifications. Our proposed approach contributes to mitigation efforts by monitoring land use and reforestation, a crucial initial stride towards expediting decarbonization and estimating greenhouse gas emissions.

2 Introduction

Since the dawn of civilization, there have been changes in land-use particularly due to two of the ancientest occupations, construction, and farming. This has an impact on our ecosystem. As the population of the world started to increase, these two professions started to have an adverse impact on our ecosystem as well. During the period between 1950 and 2015, United Nations Development Programme (UNDP) estimated that the population in the cities jumped from 54.6% to 78.3% (Un-Habitat 2016). With the stellar growth of the population of world, humans are claiming more and more land that originally belong to the forests to develop cities, industries, and farms to meet the increasing demands of residence, and food supply. Consequently, this is causing the forests to shrink day by day causing wildfires due to global warming. In this inexorable population surge, monitoring the land-use is of utmost importance.

In order to learn about future planning and geography, it is crucial to analyze construction and cultivation based on spatial data from the past and present (Dadras et al. 2015). We can gauge the disparity in land-use over time by utilizing satellite imagery via spatio-temporal analysis. Satellite imagery can account for the destruction brought about by war, deforestation, and natural disasters. For this purpose, we need to design a system that can intelligently recognize and categorize the geographical change in land-use or land-cover. Fortunately, now we can analyze even large-scale parameters worldwide because of the recent trend and development in high-resolution satellite imagery and machine learning, especially due to Deep Convolutional Neural Networks (CNNs).

Deep learning, particularly CNNs have in the recent past revolutionized many machine learning tasks. Examples include image classification (Krizhevsky, Sutskever, and Hinton 2017; Li et al. 2019; Zhang et al. 2019b), video process-



STAG-NN-BA-GCP STAG-NN-BA-GSP

Figure 1: Spatio-Temporal driven Attention Graph Neural Network with Block Adjacency matrix (STAG-NN-BA).

ing (Sharma et al. 2021; Sreenu and Durai 2019), speech recognition (Laux et al. 2023; Zhang et al. 2022), and natural language processing (Zhu et al. 2022; Lucic et al. 2022). These applications are usually characterized by data drawn from Euclidean space. However, measurements over the surface of the earth are inherently non-euclidean in nature due to its irregular and changing shape and its high mountains and deep ocean trenches. Data from such non-euclidean space can be represented as graphs (Kipf and Welling 2016; Bliss and Schmidt 2013; Velickovic et al. 2017) so as to capture the complex relationships and interdependency between objects. Recently, many studies on extending deep learning approaches for graph data have emerged (Henaff, Bruna, and LeCun 2015; Defferrard, Bresson, and Vandergheynst 2016; Jain et al. 2016; Kipf and Welling 2016; Wang et al. 2018; Satorras and Estrach 2018; Narasimhan, Lazebnik, and Schwing 2018; Hu et al. 2018; Gu et al. 2018; Wang, Ye, and Gupta 2018; Lee et al. 2018; Qi et al. 2018b; Marino, Salakhutdinov, and Gupta 2016; Kampffmeyer et al. 2019; Edwards and Xie 2016; Liu et al. 2020; Fey and Lenssen 2019; Wan et al. 2019; Qi et al. 2018a; Zhou and Chi 2019; Zhang, Campbell, and Gould 2020; Tompson et al. 2014). For instance, graph neural networks (GNNs) have been increasingly used for applications such as molecule and social network classification (Knyazev, Lin, and Mohamed 2018) and generation (Simonovsky and Komodakis 2017), 3D Mesh classification and correspondence (Fey et al. 2018), modeling behavior of dynamic interacting objects (Kipf et al. 2018), program synthesis (Allamanis, Brockschmidt, and Khademi 2017), reinforcement learning tasks (Bapst et al. 2019) and many other exciting problems.

While the utility of graph neural networks for emerging applications is promising, the complexity of graph data imposes significant challenges on many existing machine learning algorithms. For instance, in the area of image processing, the use of Graph Convolutional Networks (GCN) is still limited to a few examples only (Kampffmeyer et al. 2019; Wang, Ye, and Gupta 2018; Lee et al. 2018). By some carefully hand-crafted graph construction methods or other supervised approaches, images can be converted to structured graphs capable of processing by GCNs. In these GNNs, each pixel of an image is considered as a graph node (Edwards and Xie 2016) which is cumbersome and in many cases unnecessary. Instead of learning from raw image pixels, the use of 'superpixels' addresses this concern (Liang et al. 2016; Knyazev et al. 2019) and helps in reducing the graph size and thereby the computational complexity. The applications of Superpixels include saliency estimation (Zhu et al. 2014), optical flow estimation (Sevilla-Lara et al. 2016), object detection (Yan et al. 2015), semantic segmentation (Gadde et al. 2016), reduce input for subsequent algorithms (Fey and Lenssen 2019) and explainable AI (Ribeiro, Singh, and Guestrin 2016).

In this paper, we propose a hybrid attention method to incorporate these relational inductive biases in remote sensing data. Instead of classifying each pixel, we propose a method based on Simple Linear Iterative Clustering (SLIC) image segmentation and Graph Attention Network: GAT (Velickovic et al. 2017) to detect socio-economic indicators from remote sensing data. We first over-segment the image into superpixels. These superpixels become the nodes of our Graph Convolution Network (GCN). We then construct a region adjacency graph (RAG) where each superpixel is connected to every other adjacent superpixel in the image, enabling information to propagate globally. Finally, we classify each RAG via Spatially driven Attention Graph Neural Network (SAG-NN). We also propose an extension to our SAG-NN for spatio-temporal data named as Spatio-temporal Attention driven GNN (STAG-NN). Unlike, pixels or objects, superpixels are prone to change over time, to address this problem we propose a STAG-NN with Block diagonal Adjacency matrix (STAG-NN-BA) which enables us to incorporate both the spatial as well as temporal information in a single time-varying graph. The main novelty of this paper is the SAG-NN and STAG-NN-BA architectures for the prediction of spatio-temporal transition classes (such as construction, destruction, cultivation, and harvesting) from remote sensing data. We also show this approach incurs less computational cost compared with other deep learning methods. The details of our proposed approach, which is derived from vanilla GAT (Velickovic et al. 2017), are presented in Section 4.

In this paper, we propose a unified framework allowing to generalize geometric deep learning to remote sensing data and learn spatial and spatio-temporal features us-

ing superpixels. We improve the GAT scoring function to overcome the following shortcomings in GATv1 (Velickovic et al. 2017) and GATv2 (Brody, Alon, and Yahav 2021): 1) In GATv1, the learned layers $\mathbf{W}$ and a are applied consecutively, and thus can be collapsed into the single linear layer. 2) GATv2 (Brody, Alon, and Yahav 2021) performs best for a complete bipartite graph. We improved the graph attention scoring function by introducing the relational inductive bias in data using neighborhood features aggregation as well as the ranking of attended nodes. Our proposed approach achieves higher accuracy with less computing cost than state-of-the-art graph neural network architectures.

3 Challenges

3.1 Heterogeneity in Remote Sensing Data

While considering a large geographic area, several inherent complexities in satellite imagery make automated detection of change in land-use a challenging task. This includes, but is not limited to, i) variations in imaging sensors, ii) differences in construction design across the countries, iii) dynamic surroundings and iv) variations in luminosity, seasonal changes, and pollution levels, etc.

The heterogeneity of land surface covers, in particular, poses a major challenge for the task of spatial and spatio-temporal analysis. High resolution satellite imagery is drawing much attention from researchers due to the fine spatial details of land surface covers. Pixel-based classification methods are hardly applicable for high-resolution remote sensing images due to the high interior heterogeneity of land surface covers. The separation between spectral signatures of different land surface covers is more difficult due to the abundant details in pixel-based classification (Zhang et al. 2019a). To deal with this challenge, we are using superpixel-based classification which reduces the redundancy of the spatial features of different ground objects. Details of other challenges can be found in this paper (Nazir et al. 2020).

3.2 Representation of Images as Graphs

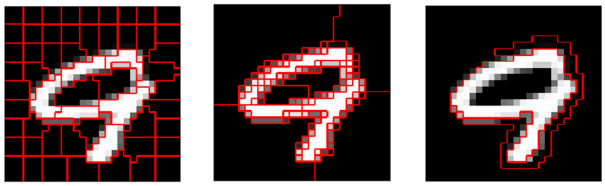
GNNs on images are characterized by unique challenges with respect to their implementation. Most of the graph neural frameworks (Defferrard, Bresson, and Vandergheynst 2016; Edwards and Xie 2016; Liu et al. 2020) are designed for dense representations such as pixel-based graphs. However, pixel based representation results in a large number of nodes which increases both the compute as well as memory costs. Since adjacent pixels are known to have similar information except at object boundaries, pixel based representation is not only cumbersome, but it is also highly redundant. To address this concern superpixel and object-based graphs have been extensively used in the literature (see Table 4 in Appendix). For subsequent processing, superpixels have been widely used as an effective way to reduce the number of image primitives.

The literature includes numerous methods for determining a superpixel based representation from an image, each with different strengths and weaknesses. Recently, many DNN-based methods to identify superpixels have been proposed (Yang et al. 2020; Jampani et al. 2018). But the most popular of practices in the GNN literature (on account of generally good results and low compute complexity) are SLIC (Achanta et al. 2012), Quickshift (Vedaldi and Soatto 2008) and Felzenszwalb (Felzenszwalb and Huttenlocher 2004). Details of these methods are presented in the following subsections.

SLIC The SLIC (simple linear iterative clustering) (Achanta et al. 2012) algorithm simply performs an iterative clustering approach in the 5D space of color information and image location. The algorithm quickly gained momentum and is now widely used due to its speed, storage efficiency, and successful segmentation in terms of color boundaries. However, the limitation of SLIC is that it often captures the background pixels as shown in Fig. 2 – Column 1, and therefore does not significantly help in data reduction for the graph generation. But it performs better in capturing built-up and grassy land from satellite imagery as shown in Fig. 5 – Column 2.

Quickshift Quickshift (Vedaldi and Soatto 2008) is a relatively recent 2D algorithm that is based on an approximation of kernelized mean-shift (Comaniciu and Meer 2002). It segments an image based on the three parameters: ϵ for the standard deviation of the Gaussian function, α for the weighting of the color term, and S to limit the calculating a window size of $S \times S$. Therefore, it belongs to the family of local mode-seeking algorithms and is applied to the 5D space consisting of color information and image location. One of the benefits of Quickshift is that it actually computes a hierarchical segmentation on multiple scales simultaneously. As shown in Fig. 2 – Column 2, it does not capture background pixels and also reduces 30% of input data for the graph generation. But it cannot segment built-up and grassy areas perfectly as shown in Fig. 5 – Column 3.

Felzenszwalb This fast 2D image segmentation algorithm, proposed in (Felzenszwalb and Huttenlocher 2004), has a single scale parameter that influences the segment size. The actual size and number of segments can vary greatly, depending on local contrast. This segmentation appeared to be less suitable in tests on a series of images, as its parameters require a special adjustment, and consequently, a static choice of this parameter leads to unusable results. As shown in Fig. 2 – Column 3 and Fig. 5 – Column 1, it only captures the pixels corresponding to the region of interest pixels but performs poorly in graph generation procedure as shown in Fig. 3 - Column 3.



SLIC Quickshift Felzenszwalb

Figure 2: Superpixel segmentation techniques on MNIST digit: 9.

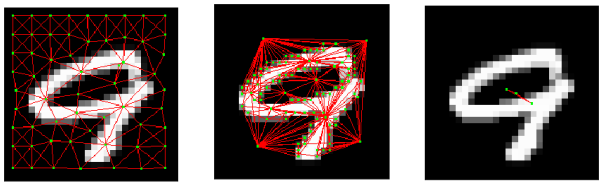
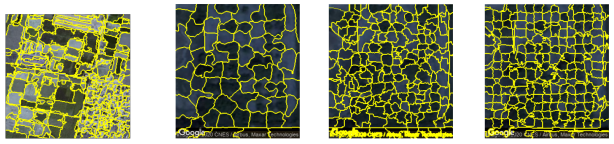
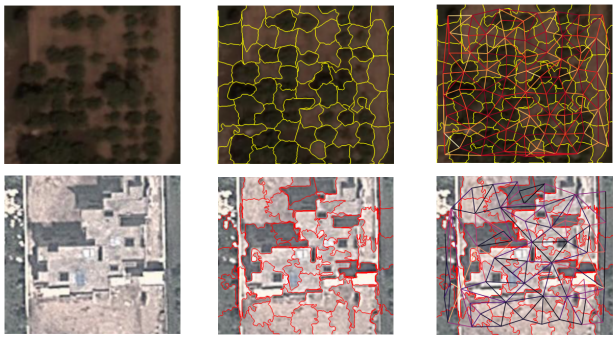


Figure 3: Region Adjacency Graphs (RAG) generation from SLIC, Quickshift and Felzenszwalb superpixels respectively.



Felzenszwalbs's SLIC Quickshift C. Watershed

Figure 4: Superpixel segmentation techniques on image from Asia14 dataset. Felzenszwalbs's method and quickshift cannot segment perfectly built-up and barren land due to inherent complexities in satellite imagery. On the other hand, compact watershed poorly performed on grassy land. While SLIC works perfectly on satellite imagery.



Satellite Image SLIC superpixels Adjacency matrix

Figure 5: RAG generation from SLIC superpixels on image from Asia14 dataset (Satellite images courtesy Google Earth).

Instead of grid-based placement as in images, superpixels usually result in irregular representation depending upon image content. Such irregular representation restricts the construction of graph on spatio-temporal data. This work has addressed this issue by proposing STAG-NN-BA which resolves the issue via a block adjacency matrix.

4 Proposed Methodology

The proposed methodology consist of following major steps:

- Generate a superpixel representation of the input images.
- Create a region adjacency graph (RAG) from the superpixel representation, by connecting neighbouring superpixels.
- Spatial Attention Graph Neural Network (SAG-NN) from region adjacency graph (RAG) for spatial classification.
- Spatio-temporal driven Graph Attention Neural Network with Block Adjacency matrix (STAG-NN-BA) for classification of transitions or changes in land-use over time.

The following subsections discuss the proposed architecture in detail.

4.1 Superpixel Segmentation

When we apply segmenation techniques on satellite imagery, SLIC (Achanta et al. 2012) performs better as compared to Quickshift (Vedaldi and Soatto 2008), Felzenszwalb (Felzenszwalb and Huttenlocher 2004) and Compact watershed (Neubert and Protzel 2014). As shown in Fig. 4 – Column 2, SLIC captures the color boundaries, and segments perfectly the built-up area and agricultural land. It is more stable for satellite imagery as compared to other segmentation techniques. The superpixel segmentation technique using SLIC (Achanta et al. 2012) provides an elegant way to divide the satellite image into homogeneous regions as shown in Fig. 4. We set the number of segments to 75 and compactness to 10. This resulted in approximately 75 superpixels per image and subsequently a graph of 75 nodes instead of 65536 nodes in case of using raw pixel values of remote sensing imagery.

4.2 Graph generation from superpixels

After using a superpixel segmentation technique, a Region Adjacency Graph (RAG) is generated by treating each superpixel as a node and adding edges between all directly adjacent superpixels. Unlike MoNet (Monti et al. 2017), which use K-Nearest Neighbours to form a connection between nodes, in our graph G we formed connections based on immediate adjacency only. Thus ours is a more compact graph while the information from neighbours of neighbours can still be incorporated in our case by using K-hop messaging passing. Each graph node can have associated features, providing aggregate information based on the characteristics of the superpixel itself. The regions obtained in the segmentation stage are represented as vertices V and relations between neighboring regions are represented as edges E. The search for the most similar pair of regions is repeated several times per iteration and every search requires $\mathcal{O}(N)$ region similarity computations. The graph is utilized so that the search is limited only to the regions that are directly connected by the graph structure.

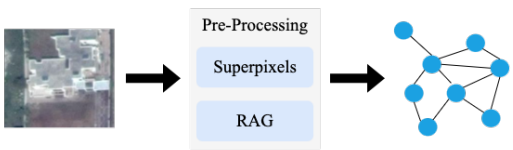


Figure 6: RAG Generation from a single geospatial image.

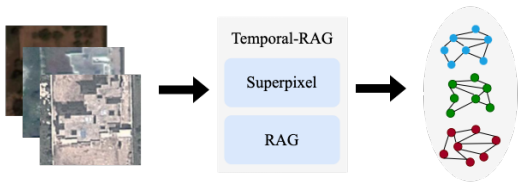


Figure 7: Generation of Temporal RAG from geospatial images of same geolocation from multiple years.

4.3 Spatial Attention Graph Neural Network (SAG-NN)

We will start by describing a single message passing layer, as the sole layer utilized throughout all of the GCN (Kipf and Welling 2016) and GAT (Velickovic et al. 2017) architectures.

Consider a graph $G(V, E)$, where V is set of n nodes and E is the set of m vertices. G is specified as a set of nodes' initial embeddings (input features): $(\overrightarrow{x_1}, \overrightarrow{x_2}, \ldots, \overrightarrow{x_n})$, and an adjacency matrix $\mathbf{ADJ}$, such that $\mathbf{ADJ}_{i,j} = 1$ if i and j are connected, and 0 otherwise. Consider node i's initial embedding (for step $k = 0$) is:

$$\overrightarrow{h}_i^{(0)} = \overrightarrow{x}_i, \forall i \in V \tag{1}$$

A graph convolutional layer at step $k = 1, 2, \ldots, K$ then computes a set of new node features $(\overrightarrow{h_1}^k, \overrightarrow{h_2}^k, \ldots, \overrightarrow{h_n}^k)$, based on the input features as well as the graph structure. Every graph convolutional layer starts off with a shared feature transformation specified by a weight matrix $\mathbf{W}$.

In general, to satisfy the localization property, we will define a graph convolutional operator as an aggregation of features across neighbourhoods; defining $\mathcal{N}_i$ as the neighbourhood of node i (typically consisting of all first-order neighbours of i, including i itself), we can define the output features of node i as

$$\overrightarrow{h}_i^{(k)} = f^{(k)}\left(\mathbf{W}^{(k)} \cdot \left[\sum_{j \in \mathcal{N}_i} C^{(k)} \overrightarrow{h}_j^{(k-1)} + C^{(k)} \overrightarrow{h}_i^{(k-1)}\right]\right) \tag{2}$$

where $\forall i \in V$ and $f^{(k)}$ is an activation function. Each neighbour can be assigned different importance as:

$$\overrightarrow{h}_i^{(k)} = f^{(k)}\left(\mathbf{W}^{(k)} \cdot \left[\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(k-1)} \overrightarrow{h}_j^{(k-1)} + \alpha_{ii}^{(k-1)} \overrightarrow{h}_i^{(k-1)}\right]\right) \tag{3}$$

where $\forall i \in V$ and $\sum_{j \in \mathcal{N}_i}(.)$ is the weighted mean of i's neighbour's embedding at step $k-1$ and the attention weights $\alpha^{(k)}$ are generated by an attention mechanism $\mathbf{A}^{(k)}$, normalized such that the sum over all neighbours of each node i is 1:

$$\alpha_{ij}^{(k)} = \frac{\mathbf{A}^{(k)}(\overrightarrow{h}_i^{(k)}, \overrightarrow{h}_j^{(k)})}{\sum_{w \in \mathcal{N}_i} \mathbf{A}^{(k)}(\overrightarrow{h}_i^{(k)}, \overrightarrow{h}_w^{(k)})}, \forall (i,j) \in E \tag{4}$$

In standard GAT (see eq. 3 & 4) α_{ij} is implicitly defined, employing self-attention over the node features to do so. This choice was not without motivation, as self-attention has previously been shown to be self-sufficient for state-of-the-art-level results on machine translation, as demonstrated by the Transformer architecture (Vaswani et al. 2017).

Generally, we let α_{ij} be computed as a byproduct of an attentional mechanism, $a : \mathcal{R}^N \times \mathcal{R}^N \longrightarrow \mathcal{R}$ which computes normalized coefficients α_{ij} across pairs of nodes i, j, based on their features (see eq. 4).

In contrast, in GATv2, every node can attend to any other node using scoring function shown in eq. 5.

$$\overrightarrow{h}_i^{(k)} = \alpha_{ij}^{(k-1)}\left[f^{(k)}\left(\mathbf{W}^{(k)} \cdot \sum_{j \in \mathcal{N}_i} \overrightarrow{h}_j^{(k-1)} + \overrightarrow{h}_i^{(k-1)}\right)\right] \tag{5}$$

The main problem in the standard GAT scoring function (see eq. 3) is that the learned layers $\mathbf{W}$ and α are applied consecutively, and thus can be collapsed into single linear layer (Brody, Alon, and Yahav 2021). To fix this limitation in our work, we then impose a relational inductive bias in data using neighborhood features aggregation (see eq. 6 & 7). In our proposed SAG-NN, the node i's embedding at step k for $k = 1$ is:

$$\overrightarrow{h}_i^{(k)} = f^{(k)}\left(\mathbf{W}^{(k)} \cdot \left[AGG_{j \in \mathcal{N}_i}(\{\overrightarrow{h}_j^{(k-1)}\}), \overrightarrow{h}_i^{(k-1)}\right]\right), \tag{6}$$

where $\forall i \in V$ and $AGG(.)$ is the aggregation of i's neighbour's embeddings at step $k-1$ and $h_i^{(k-1)}$ is the node i's embedding at step $k-1$. And node i's embedding at step k for $k = 2, 3, \ldots$ upto K is:

$$\overrightarrow{h}_i^{(k)} = f^{(k)}\left(\mathbf{W}^{(k)} \cdot \left[\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(k-1)} \overrightarrow{h}_j^{(k-1)} + \alpha_{ii}^{(k-1)} \overrightarrow{h}_i^{(k-1)}\right]\right) \tag{7}$$

The proposed solution not only improves the aggregation of features from neighbouring nodes, it also improves the ranking of attended nodes (static attention) as shown in eq. 6 & 7.

Spatio-temporal Classification via SAG-NN-E Although the proposed SAG-NN architecture is developed to account for neighborhood features' aggregation to learn spatial land-use classes, we also extended it for spatio-temporal classification. Given T time steps, our resulting ensemble SAG-NN-E has T copies of SAG-NN, one for each time step, connected in parallel. The ensemble has a voting scheme that takes the spatial classification from each SAG-NN and generates the spatio-temporal classification (see Fig. 8). We used this ensemble as a baseline for evaluation of our proposed Spatio-temporal driven Graph Attention Neural Network which is discussed next.

4.4 Spatio-temporal driven Graph Attention Neural Network with Block Adjacency matrix (STAG-NN-BA)

Images having multiple channels such as in case of color or multi-spectral images or sequence of multiple images are usually represented as a spatio-temporal volume. These

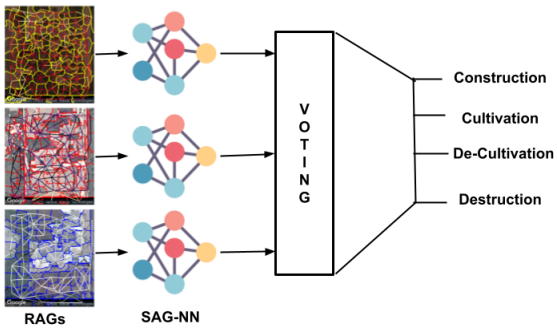


Figure 8: Spatio-temporal Classificcation via SAG-NN-E.

patio-temporal volumes have fixed spatial dimension or pixels at each depth of the volume. However, when instead of pixels, superpixels of images are used this result in different dimension at each time step. Thus graph from superpixels of each image from a sequence cannot be stacked together as in case of pixel based representation. Furthermore, in GNNs the structure of the graph remains unchanged over multiple layers, only the node representation changes (Kipf and Welling 2016). This restricts the use of GNNs for spatio-temporal classification problems having varying nodes over time.

We addressed this problem by proposing a novel temporal-RAG that connects the individual RAG from each image. To incorporate the temporal change in graphs, we add the fourth dimension in the node features of these RAGs which is basically a numeric index that indicates the chronological order of the image the superpixel belongs to. We then combine the RAGs of these separate images into a supergraph that has these RAGs as unconnected subgraphs, we call this supergraph *Temporal-RAGs*. Figure 7 depicts the creation of *Temporal-RAGs* from Images of a geo-location from different years. Our proposed temporal-RAG is an extension of our SAG-NN architecture. The supergraph of SAG-NN's is generated by combining the adjacency matrices from each RAG into a single adjacency matrix (see Figs. 7). This results in a block diagonal adjacency matrix for Temporal-RAGs resulting in Spatio-temporal driven Graph Attention Neural Network with Block Adjacency matrix (STAG-NN-BA) defined as:

$$\begin{aligned}\overrightarrow{h}_i^{(k)} = ReLU\Big(\mathbf{W}^{(k)}.\Big[\sum_{j\in\mathcal{N}_i}\alpha_{ij}^{(k-1)}\overrightarrow{h}_j^{(k-1)} + \alpha_{ii}^{(k-1)}\overrightarrow{h}_i^{(k-1)}\Big]\Big) \\ +\!\!+\, ReLU\Big(\mathbf{W}^{(k)}.\Big[\sum_{j\in\mathcal{N}_i}\alpha_{ij}^{(k-1)}\overrightarrow{h}_j^{(k-1)} + \alpha_{ii}^{(k-1)}\overrightarrow{h}_i^{(k-1)}\Big]\Big) \\ +\!\!+\, ReLU\Big(\mathbf{W}^{(k)}.\Big[\sum_{j\in\mathcal{N}_i}\alpha_{ij}^{(k-1)}\overrightarrow{h}_j^{(k-1)} + \alpha_{ii}^{(k-1)}\overrightarrow{h}_i^{(k-1)}\Big]\Big)\end{aligned} \tag{8}$$

where $+\!\!+$ symbol represent the concatenation of features.

In STAG-NN-BA we aggregate the node embedding from all the RAGs into one graph embedding X_G of length D. Then, we feed that embedding to the Multi-Layer Perceptron (MLP) for assigning one of the final transition classes. Our proposed architecture allows to impose relational inductive bias in data using neighborhood features aggregation over space as well as time resulting in a single architecture for data with a varying number of nodes over time (see Fig. 1). Thus it can be used to classify the transitions or change in land-use over time in the remote sensing data. Since transitions are essentially temporal phenomena, the proposed STAG-NN-BA method can incorporate temporal information into regional adjacency graphs. We believe that this method can be extended to other geometric data.

We do not assign features to the edges, since our model uses an attention mechanism, and we believe that the edge features will be learned according to the features of the connecting nodes. STAG-NN-BA combine ideas of graph convolutions (Kipf and Welling 2016), which allows graph nodes to aggregate information from their irregular neighbourhoods, with self-attention mechanisms (Vaswani et al. 2017), which allows nodes to learn the relative importance of each neighbour during the aggregation process.

Although, there are many different models that try to incorporate weights in neighborhood aggregation such as SplineCNN (Fey et al. 2018) and GEO-GCN (Spurek et al. 2019). We used three approaches to perform a land-use transition classification of temporal images namely SAG-NN-E (see section 4.3), Global Sum Pooling (STAG-NN-BA-GSP) and Global Concatenated Pooling (STAG-NN-BA-GCP). The last two are discussed as follows:

Global Sum Pooling (STAG-NN-BA-GSP): There exist many different types of order-in-variant read-out layers in the literature, such as Global Average Pooling (Lin, Chen, and Yan 2013), Global Attention Pooling (Li et al. 2015), Global Max Pooling (Lin, Chen, and Yan 2013), and Global Sum Pooling (Li et al. 2015).

We use Global Sum Pooling (GSP) for it's simplicity as defined in the equation: $\mathbf{x}_{\mathcal{G}} = \sum_{v\in\mathcal{V}}\mathbf{x}_v^{(L)}$, where V is the set of vertices, $\mathbf{x}_v^{(L)}$ is the node embedding at the last layer of a graph neural network, and $\mathbf{x}_{\mathcal{G}}$ is the embedding for the graph obtained as a result of the pooling operation.

Global Concatenated Pooling (STAG-NN-BA-GCP): We are using RAGs of images from three different timestamps combined into one Temporal-RAGs for the transition classification. Taking the graph readout in the last layers of GAT using Global Sum Pooling (GSP) adding all the nodes of the Temporal-RAGs into one n-dimensional vector. This makes the embedding of a Temporal-RAG indistinguishable from the embedding of a Temporal-RAG in which the underlying RAGs were to swap places. To solve this problem, we introduced a variation of GSP which gives us separate embedding for each underlying RAG concatenated into one $n \times D$ vector (see Fig. 1).

5 Results and Evaluation

5.1 Datasets

We used three datasets for evaluation of our proposed approch namely MNIST (LeCun et al. 1998), Asia14 (Nazir et al. 2020) and C2D2 Dataset (Bhimra, Nazir, and Taj 2019). Both Asia14 and C2D2 datasets are remote sensing

Table 1: Spatial classification accuracy on pixel based Region Adjacency Graph (RAG) of MNIST (LeCun 1998) and subset of Asia14 (Nazir et al. 2020) datasets. Top-2 ranking methods are in bold and, in particular, red (1st) and violet (2nd).

Architectures	#Param (M)	MNIST	Asia14
Classical models of neural network on image dataset			
Inception-ResNet-v2 (Szegedy et al. 2017)	23.50	-	57.70 %
2D-ResNet-50 (He et al. 2016)	23.50	-	56.45 %
Graph neural networks			
MoNet (Monti et al. 2017)	**2.12**	91.11%	66.39%
ChebNet (Defferrard, Bresson, and Vandergheynst 2016)	12.85	75.62%	64.60 %
GATv1 (Velickovic et al. 2017)	25.70	96.19%	69.85 %
AGNN (Thekumparampil et al. 2018)	**0.41**	97.98%	47.80%
GraphSAGE (Hamilton, Ying, and Leskovec 2017)	12.85	97.27%	70.00%
Crystal GCN (Xie and Grossman 2018)	**0.41**	**98.04%**	63.20%
GATv2 (Brody, Alon, and Yahav 2021)	25.70	-	**71.10%**
SAG-NN (our)	25.69	**98.14%**	**77.00%**

Table 2: Spatial classification accuracy on SLIC superpixels based Region Adjacency Graph (RAG) of subset of Asia14 (Nazir et al. 2020) datasets. Top-2 ranking methods are in bold and, in particular, red (1st) and violet (2nd).

Architectures	#Param (M)	Asia14
Classical models of neural network on image dataset		
Inception-ResNet-v2 (Szegedy et al. 2017)	23.50	57.70 %
2D-ResNet-50 (He et al. 2016)	23.50	56.45 %
Graph neural networks		
GCN (Kipf and Welling 2016)	**0.015**	9.78%
GraphSAGE (Hamilton, Ying, and Leskovec 2017)	**0.015**	65.00%
GATv1 (Velickovic et al. 2017)	**0.030**	**80.30**
GATv2 (Brody, Alon, and Yahav 2021)	0.055	72.04%
SAG-NN (our)	**0.030**	**80.98%**

datasets for spatial and spatio-temporal classification respectively. These datasets are graph signal classification tasks, where graphs are represented in mixed mode: one adjacency matrix, many instances of node features. Details of these datasets are discussed next.

MNIST Pixel-based Dataset The MNIST dataset (LeCun et al. 1998) is an acronym that stands for the Modified National Institute of Standards and Technology dataset. It is a dataset of 28×28 pixel grayscale images of handwritten single digits between 0 and 9. MNIST dataset containing $70,000$ pixel based region adjacency graphs as described by (Defferrard, Bresson, and Vandergheynst 2016). Every graph is labeled by one of 10 classes.

Asia14 pixel-based and Superpixels Dataset *Asia14* dataset contains samples under varying conditions as discussed in Section 3.1. Furthermore, unlike street imagery, land-use is subject to significant variations in satellite imagery. To cater for this, we used a subset of 14-class dataset named *Asia14* (Nazir et al. 2020). This dataset consisting of Digital Globe RGB band images from 2016 and 2017 of resolution 256×256 at zoom level 20 (corresponding to 0.149 pixel per meter on the equator). We used 9 classes including brick kilns, houses, roads, tennis courts, grass, dense forest, parking lots, parks. The issue of sensor variations is handled by diversifying the training data across several spatial locations within the Indo-Pak region of South Asia. We have $9,000$ pixel-based region adjacency graphs and we generated the superpixels using SLIC (Achanta et al. 2012). Then $9,000$ graphs with 75 nodes each, were generated using region adjacency graph method.

C2D2 Dataset This dataset contains Spatio-temporal data annotated for four fundamental land-use land-change transitions namely construction, destruction, cultivation, and decultivation. This dataset was originally collected and prepared by (Bhimra, Nazir, and Taj 2019). They browsed Digital Globe imagery data for the years 2011, 2013, and 2017 and visited almost $5,50,000$ random locations which make approximately 5310 km^2. Along with lat-long, at each location, we cropped an image patch of resolution 256×256 at zoom level 20 (i.e. 0.149 pixel per meter on the equator). The provided dataset contained 3D volumes of Spatio-temporal images from different years. we had to reverse the process to separate out the individual images for a location into the directories of each year. We then generate regional adjacency graphs (RAG)s from the superpixels of these images that were generated using SLIC and use the same annotations as it was assigned to the 3D volumes.

5.2 Evaluation of SAG-NN

We evaluated our Spatial Attention Graph Attention Network (SAG-NN) architecture on two datasets namely MNIST and Asia14. We performed two experiments, in the first experiment we generated pixel-based graphs and in the

Table 3: Spatio-temporal comparative evaluation for land-use transition classification on C2D2 dataset respectively. (Key: Acc.: Accuracy, Par.: Parameters, M: Millions, FPT: Forward Pass Time in milliseconds for 100 forward passes). Top-2 ranking methods are in bold and, in particular, red (1st) and violet (2nd).

Model	# Par. (M)	FPT (ms)	Acc.
3D-ResNet-34 (Bhimra, Nazir, and Taj 2019)	63.50	¿ 3.6	57.72 %
SAG-NN-E	**0.030**	3.60 ms	60.02 %
STAG-NN-BA-GCP (ours)	**0.050**	**2.50 ms**	**64.90 %**
STAG-NN-BA-GSP (ours)	**0.030**	**2.62 ms**	**77.83 %**

second experiment we used superpixel based graphs. We performed comparisons with two classical methods namely Inception-ResNet-v2 (Szegedy et al. 2017) and 2D-ResNet-50 (He et al. 2016) and seven graph based state-of-the-art methods namely MoNet (Monti et al. 2017), ChebNet (Defferrard, Bresson, and Vandergheynst 2016), GATv1 (Velickovic et al. 2017) , AGNN (Thekumparampil et al. 2018), GraphSAGE (Hamilton, Ying, and Leskovec 2017), Crystal GCN (Xie and Grossman 2018), GATv2 (Brody, Alon, and Yahav 2021).

We first trained and validated our Spatial Attention Graph Attention as well as all the other methods on MNIST dataset. Our SAG-NN model achieved highest accuracy of 98.14% on MNIST dataset with 25.69 million number of parameters on pixel-based RAG. Then we trained and tested SAG-NN as well as all the other methods on Asia14 dataset. Here again our proposed SAG-NN achieved highest test accuracy of 77.00% and 80.98% on pixel-based graph and superpixel RAGs respectively (see Table 1 and 2).

In Table 1, the experiments show that the SAG-NN outperforms on pixel-based RAGs as compared to other classical or RAG-based GNN classifiers. In Table 2, SAG-NN has comparable training parameters and shows high accuracy when compared with GCN (Kipf and Welling 2016) and GraphSAGE (Hamilton, Ying, and Leskovec 2017). It shows comparable high accuracy when compared with GATv1 (Velickovic et al. 2017). GATv2 (Brody, Alon, and Yahav 2021) is proposed for bipartite graphs that's why it shows low performance on pixel-based and superpixel-based region adjacency graphs as compared to our proposed model.

5.3 Evaluation of STAG-NN-BA

We compared both the variants of our STAG-NN-BA with two other methods namely 3D-ResNet-34 (Bhimra, Nazir, and Taj 2019) and SAG-NN-E. SAG-NN-E is our extension of SAG-NN for spatio-temporal data and serves as the baseline. 3D-ResNet-34 (Bhimra, Nazir, and Taj 2019) on the other hand uses 3D convolution and is the only state-of-the-art method with published results on C2D2 dataset. In order to compare our results on C2D2 dataset, we used the same train/test split as in 3D-ResNet-34 (Bhimra, Nazir, and Taj 2019). The ability of transition classification for SAG-NN-E approach is dependent on the performance of land-use classification and voting procedure (see section 4.3). Both STAG-NN-BA-GCP and STAG-NN-BA-GSP achieved significantly higher accuracies as compared to SAG-NN-E and 3D-ResNet-34 (Bhimra, Nazir, and Taj 2019) in terms of accuracy and compute cost. STAG-NN-BA-GCP and STAG-NN-BA-GSP achieved approximately 7% and 20% higher accuracy as compared to 3D-ResNet-34. They also achieved 4.88% and 17.81% higher accuracy as compared to SAG-NN-E which indicates the effectiveness of our temporal model STAG-NN-BA as compared to spatial model via SAG-NN. Furthermore, STAG-NN-BA-GSP ourperforms all the other methods which shows that the global sum pooling is a more suited method of aggregation as compared to global concatenated pooling.

Table 3 also compares the training parameters, forward pass time, and accuracy of our models used for spatio-temporal land-use classification. It can be seen that the forward pass time of STAG-NN-BA is almost 1ms lower as compared to SAG-NN and much lower as compared to 3D-ResNet-34.

In the land-use transition classification, the STAG-NN-BA-GSP approach is the most reliable. However, we also draw comparison of 3D-ResNet-34 (Bhimra, Nazir, and Taj 2019) with SAG-NN-E and STAG-NN-BA-GCP (see Table 3). Both spatio-temporal proposed models (STAG-NN-BA-GCP and STAG-NN-BA-GSP) achieved higher performance with low computational cost on the C2D2 dataset.

6 Conclusion and Future work

This paper proposed two novel Graph Neural Network architectures for spatial and spatio-temporal classification of remote sensing imagery to gain a deeper understanding of socio-economic indicators. We also proposed a novel method to represent temporal information in images using region adjacency graph called Temporal-RAG. We evaluated our approaches on two remote sensing datasets namely Asia14 and C2D2. The comparison with the previously existing classical and graph neural network methods showed that our approaches achieved higher performance and reduced the computation power greatly. There are two areas recognized while working on this paper that can serve as interesting problems for future works. Firstly, there is an issue of information loss during the generation of graphs from superpixel segmentation. Secondly, over-segmentation of an image to make superpixels causes information loss, which decreases the representation power of pixels-based graphs. The information about the shape of the underlying superpixel segment is lost. We can extract generic shape embedding using an auto-encoder into a single N dimensional vector. While assigning the color values as features, this N-dimensional shape embedding vector can be concatenated into the initial features. This can help incorporate the shape

into graph representations.

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Süsstrunk, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11): 2274–2282.

Allamanis, M.; Brockschmidt, M.; and Khademi, M. 2017. Learning to represent programs with graphs. *arXiv preprint arXiv:1711.00740*.

Bapst, V.; Sanchez-Gonzalez, A.; Doersch, C.; Stachenfeld, K.; Kohli, P.; Battaglia, P.; and Hamrick, J. 2019. Structured agents for physical construction. In *International Conference on Machine Learning*, 464–474. PMLR.

Bhimra, M. A.; Nazir, U.; and Taj, M. 2019. Using 3D Residual Network for Spatio-temporal Analysis of Remote Sensing Data. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1403–1407. IEEE.

Bliss, N. T.; and Schmidt, M. C. 2013. Confronting the challenges of graphs and networks. *Lincoln laboratory journal*, 20(1).

Blumberg, D.; and Jacobson, D. 1997. New frontiers: remote sensing in social science research. *The American Sociologist*, 28(3): 62–68.

Bogo, F.; Romero, J.; Loper, M.; and Black, M. J. 2014. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3794–3801.

Boyd, D. S.; Jackson, B.; Wardlaw, J.; Foody, G. M.; Marsh, S.; and Bales, K. 2018. Slavery from space: Demonstrating the role for satellite remote sensing to inform evidence-based action related to UN SDG number 8. *ISPRS journal of photogrammetry and remote sensing*, 142: 380–388.

Brody, S.; Alon, U.; and Yahav, E. 2021. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*.

Bronstein, M. M.; Bruna, J.; LeCun, Y.; Szlam, A.; and Vandergheynst, P. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4): 18–42.

Comaniciu, D.; and Meer, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5): 603–619.

Dadras, M.; Shafri, H. Z.; Ahmad, N.; Pradhan, B.; and Safarpour, S. 2015. Spatio-temporal analysis of urban growth from remote sensing data in Bandar Abbas city, Iran. *The Egyptian Journal of Remote Sensing and Space Science*, 18(1): 35–52.

Dai, Z.; Liu, H.; Le, Q.; and Tan, M. 2021. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34.

Danel, T.; Spurek, P.; Tabor, J.; Śmieja, M.; Struski, Ł.; Słowik, A.; and Maziarka, Ł. 2020. Spatial graph convolutional networks. In *International Conference on Neural Information Processing*, 668–675. Springer.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, 3844–3852.

Eder, M.; and Frahm, J.-M. 2019. Convolutions on spherical images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 1–5.

Edwards, M.; and Xie, X. 2016. Graph based convolutional neural network. *arXiv preprint arXiv:1609.08965*.

Felzenszwalb, P. F.; and Huttenlocher, D. P. 2004. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2): 167–181.

Fey, M.; and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*.

Fey, M.; Lenssen, J. E.; Weichert, F.; and Müller, H. 2018. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 869–877.

Gadde, R.; Jampani, V.; Kiefel, M.; Kappler, D.; and Gehler, P. V. 2016. Superpixel convolutional networks using bilateral inceptions. In *European conference on computer vision*, 597–613. Springer.

Gu, J.; Hu, H.; Wang, L.; Wei, Y.; and Dai, J. 2018. Learning region features for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 381–395.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 1024–1034.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Henaff, M.; Bruna, J.; and LeCun, Y. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*.

Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; and Wei, Y. 2018. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3588–3597.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Huo, Y.; Xu, Z.; Bao, S.; Assad, A.; Abramson, R. G.; and Landman, B. A. 2018. Adversarial synthesis learning enables segmentation without target modality ground truth. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 1217–1220. IEEE.

Jain, A.; Zamir, A. R.; Savarese, S.; and Saxena, A. 2016. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 5308–5317.

Jampani, V.; Sun, D.; Liu, M.-Y.; Yang, M.-H.; and Kautz, J. 2018. Superpixel sampling networks. In *Proceedings*

of the European Conference on Computer Vision (ECCV), 352–368.

Jiang, B.; Lin, D.; Tang, J.; and Luo, B. 2019. Data Representation and Learning With Graph Diffusion-Embedding Networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10406–10415.

Jiang, Y.; Koppula, H.; and Saxena, A. 2013. Hallucinated humans as the hidden context for labeling 3d scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2993–3000.

Kampffmeyer, M.; Chen, Y.; Liang, X.; Wang, H.; Zhang, Y.; and Xing, E. P. 2019. Rethinking knowledge graph propagation for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11487–11496.

Kipf, T.; Fetaya, E.; Wang, K.-C.; Welling, M.; and Zemel, R. 2018. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, 2688–2697. PMLR.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Knyazev, B.; Lin, X.; Amer, M. R.; and Taylor, G. W. 2019. Image classification with hierarchical multigraph networks. *arXiv preprint arXiv:1907.09000*.

Knyazev, B.; Lin, X.; and Mohamed, R. 2018. Amer, and Graham W Taylor. Spectral multigraph networks for discovering and fusing relationships in molecules. In *NeurIPS Workshop on Machine Learning for Molecules and Materials*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.

Laux, H.; Hallawa, A.; Assis, J. C. S.; Schmeink, A.; Martin, L.; and Peine, A. 2023. Two-stage visual speech recognition for intensive care patients. *Scientific Reports*, 13(1): 928.

LeCun, Y. 1998. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature*, 521(7553): 436.

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

Lee, C.-W.; Fang, W.; Yeh, C.-K.; and Frank Wang, Y.-C. 2018. Multi-label zero-shot learning with structured knowledge graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1576–1585.

Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; and Benediktsson, J. A. 2019. Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9): 6690–6709.

Li, Y.; Tarlow, D.; Brockschmidt, M.; and Zemel, R. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*.

Liang, X.; Shen, X.; Feng, J.; Lin, L.; and Yan, S. 2016. Semantic object parsing with graph lstm. In *European Conference on Computer Vision*, 125–143. Springer.

Lin, M.; Chen, Q.; and Yan, S. 2013. Network in network. *arXiv preprint arXiv:1312.4400*.

Liu, Q.; Xiao, L.; Yang, J.; and Wei, Z. 2020. CNN-Enhanced Graph Convolutional Network With Pixel-and Superpixel-Level Feature Fusion for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*.

Lucic, A.; Bleeker, M.; Bhargav, S.; Forde, J.; Sinha, K.; Dodge, J.; Luccioni, S.; and Stojnic, R. 2022. Towards reproducible machine learning research in natural language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 7–11.

Marino, K.; Salakhutdinov, R.; and Gupta, A. 2016. The more you know: Using knowledge graphs for image classification. *arXiv preprint arXiv:1612.04844*.

Mehmood, M.; Shahzad, A.; Zafar, B.; Shabbir, A.; and Ali, N. 2022. Remote sensing image classification: A comprehensive review and applications. *Mathematical Problems in Engineering*, 2022: 1–24.

Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; and Bronstein, M. M. 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5115–5124.

Narasimhan, M.; Lazebnik, S.; and Schwing, A. 2018. Out of the box: Reasoning with graph convolution nets for factual visual question answering. In *Advances in Neural Information Processing Systems*, 2654–2665.

Nazir, U.; Mian, U. K.; Sohail, M. U.; Taj, M.; and Uppal, M. 2020. Kiln-net: A gated neural network for detection of brick kilns in South Asia. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13: 3251–3262.

Neubert, P.; and Protzel, P. 2014. Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. In *2014 22nd international conference on pattern recognition*, 996–1001. IEEE.

Prakash, D. C.; Narayanan, R.; Ganesh, N.; Ramachandran, M.; Chinnasami, S.; and Rajeshwari, R. 2022. A study on image processing with data analysis. In *AIP Conference Proceedings*, volume 2393, 020225. AIP Publishing LLC.

Qi, M.; Qin, J.; Li, A.; Wang, Y.; Luo, J.; and Van Gool, L. 2018a. stagnet: An attentive semantic rnn for group activity recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 101–117.

Qi, S.; Wang, W.; Jia, B.; Shen, J.; and Zhu, S.-C. 2018b. Learning human-object interactions by graph parsing neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 401–417.

Raei, E.; Asanjan, A. A.; Nikoo, M. R.; Sadegh, M.; Pourshahabi, S.; and Adamowski, J. F. 2022. A deep learning image segmentation model for agricultural irrigation system classification. *Computers and Electronics in Agriculture*, 198: 106977.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Satorras, V. G.; and Estrach, J. B. 2018. Few-shot learning with graph neural networks.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Sevilla-Lara, L.; Sun, D.; Jampani, V.; and Black, M. J. 2016. Optical flow with semantic segmentation and localized layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3889–3898.

Sharma, V.; Gupta, M.; Kumar, A.; and Mishra, D. 2021. Video processing using deep learning techniques: A systematic literature review. *IEEE Access*, 9: 139489–139507.

Simonovsky, M.; and Komodakis, N. 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3693–3702.

Spurek, P.; Danel, T.; Tabor, J.; Smieja, M.; Struski, L.; Slowik, A.; and Maziarka, L. 2019. Geometric graph convolutional neural networks. *arXiv preprint arXiv:1909.05310*.

Sreenu, G.; and Durai, S. 2019. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *Journal of Big Data*, 6(1): 1–27.

Stutz, D.; Hermans, A.; and Leibe, B. 2018. Superpixels: An evaluation of the state-of-the-art. *ArXiv*, abs/1612.01601.

Su, Y.-C.; and Grauman, K. 2019. Kernel Transformer Networks for Compact Spherical Convolution. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9434–9443.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. 2017. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Association for the Advancement of Artificial Intelligence*, volume 4, 12.

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

Thekumparampil, K. K.; Wang, C.; Oh, S.; and Li, L.-J. 2018. Attention-based graph neural network for semi-supervised learning. *arXiv preprint arXiv:1803.03735*.

Tompson, J.; Jain, A.; LeCun, Y.; and Bregler, C. 2014. Joint training of a convolutional network and a graphical model for human pose estimation. *arXiv preprint arXiv:1406.2984*.

Un-Habitat. 2016. World cities report 2016: Urbanization and development–emerging futures. *United Nations Human Settlements Programme*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Vedaldi, A.; and Soatto, S. 2008. Quick shift and kernel methods for mode seeking. In *European conference on computer vision*, 705–718. Springer.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *stat*, 1050: 20.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio', P.; and Bengio, Y. 2018. Graph Attention Networks. *ArXiv*, abs/1710.10903.

Wan, S.; Gong, C.; Zhong, P.; Du, B.; Zhang, L.; and Yang, J. 2019. Multiscale dynamic graph convolutional network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5): 3162–3177.

Wang, X.; Ye, Y.; and Gupta, A. 2018. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6857–6866.

Wang, Z.; Chen, T.; Ren, J.; Yu, W.; Cheng, H.; and Lin, L. 2018. Deep reasoning with knowledge graph for social relationship understanding. *arXiv preprint arXiv:1807.00504*.

Xie, M.; Jean, N.; Burke, M.; Lobell, D.; and Ermon, S. 2016. Transfer learning from deep features for remote sensing and poverty mapping. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Xie, T.; and Grossman, J. C. 2018. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters*, 120(14): 145301.

Yan, J.; Yu, Y.; Zhu, X.; Lei, Z.; and Li, S. Z. 2015. Object detection by labeling superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5107–5116.

Yang, F.; Sun, Q.; Jin, H.; and Zhou, Z. 2020. Superpixel segmentation with fully convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13964–13973.

Zhai, X.; Kolesnikov, A.; Houlsby, N.; and Beyer, L. 2021. Scaling Vision Transformers. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1204–1213.

Zhang, A.; Zhang, S.; Sun, G.; Li, F.; Fu, H.; Zhao, Y.; Huang, H.; Cheng, J.; and Wang, Z. 2019a. Mapping of coastal cities using optimized spectral–spatial features based multi-scale superpixel classification. *Remote Sensing*, 11(9): 998.

Zhang, B.; Wu, D.; Peng, Z.; Song, X.; Yao, Z.; Lv, H.; Xie, L.; Yang, C.; Pan, F.; and Niu, J. 2022. Wenet 2.0: More productive end-to-end speech recognition toolkit. *arXiv preprint arXiv:2203.15455*.

Zhang, F. Z.; Campbell, D.; and Gould, S. 2020. Spatio-attentive Graphs for Human-Object Interaction Detection. *arXiv preprint arXiv:2012.06060*.

Zhang, J.; Xie, Y.; Wu, Q.; and Xia, Y. 2019b. Medical image classification using synergic deep learning. *Medical image analysis*, 54: 10–19.

Zhou, P.; and Chi, M. 2019. Relation parsing neural network for human-object interaction detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 843–851.

Zhu, C.; Xu, Y.; Ren, X.; Lin, B.; Jiang, M.; and Yu, W. 2022. Knowledge-augmented methods for natural language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 12–20.

Zhu, W.; Liang, S.; Wei, Y.; and Sun, J. 2014. Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2814–2821.

A Survey of Relevant Literature

Image Classification: Availability of high resolution satellite imagery paved a way for future planning and geographical studies for large-scale analysis across the globe (Boyd et al. 2018; Blumberg and Jacobson 1997; Huo et al. 2018; LeCun, Bengio, and Hinton 2015; Xie et al. 2016; Nazir et al. 2020). Automated large-scale surveys via remote sensing often make use of image classification (Mehmood et al. 2022) and segmentation (Prakash et al. 2022; Raei et al. 2022). In this study, we are focusing on image classification as it plays an important role in land-use land-cover applications. The generic problem of image classification consists of distinguishing the images into object classes (usually, a set of predefined collection of labels). Traditional approaches followed the method of preprocessing images to extract image features (e.g., texture, color, etc.) and running a classifier on those features.

Krizhevsky et al.(Krizhevsky, Sutskever, and Hinton 2012) published a seminal study that explored deep neural networks for image classification. They won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 by a large margin and set a turning point for image classification research. In the following years networks like GoogLeNet (Szegedy et al. 2015) and Squeeze-and-Excitation (Hu, Shen, and Sun 2018) further helped in reducing the top-5 error rate from 15.3% to just 2.51%. More recent approaches, such as ViT-e (Zhai et al. 2021) and CoAtNet-7 (Dai et al. 2021), combined convolution with attention /transformers (Vaswani et al. 2017) and achieved a top-1 accuracy of 90.45% and 90.88% respectively.

Despite the recent advances in datasets and network architectures, using standard convolutional kernels limits the applications of these networks in problems that do not present a domain based on rectangular grids. For example, panoramas capture a whole 360-degree field of view, similalry, measuerment on earth's surface are geodesic in nature. To handle these issues, some researchers suggested networks designed to adapt to the spherical domain (Eder and Frahm 2019). In contrast, others proposed to learn how to adapt convolutional layers to the spherical domain (Su and Grauman 2019). More recently, graph based methods have been introduced so that such non-Euclidean spaces can be modeled via geometric deep learning (Monti et al. 2017).

Geometric Deep Learning: Recently, there has been an increasing interest in geometric deep learning (Monti et al. 2017), attempting to generalize deep neural models to non-Euclidean structured domains such as graphs and manifolds. Graph-based representations can be used to model a variety of problems and domains. Some examples include social networks in computational social sciences, sensor networks in communications, functional networks in brain imaging, regulatory networks in genetics, and meshed surfaces in computer graphics (Bronstein et al. 2017). In addition, they naturally allow to model multi-resolution representations of the same object. Furthermore, they naturally allow several "multi-resolution" representations of the same object. The same image can be converted to graphs using pixel-level or superpixel-level representations. Superpixel-based representations reduce the input size while also allowing domains

such as pinhole and spherical images to be represented as graphs, reducing computation costs needed for classification. Furthermore, there are several recent advances toward the development of Graph Neural Networks (GNNs) (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2016; Velickovic et al. 2017), including Graph Attention Networks (Velickovic et al. 2017), which could bridge the gap between different domains.

Figure 9: Examples showing the change in land-use between 2011 and 2017. In all three examples, more and more land was used for construction purposes over the years. See Section C for a discussion on results. (Satellite images courtesy Google Earth).

Image Classification via Graphs: To the best of our knowledge, Monti et al.(Danel et al. 2020) proposed the first application of Graph Neural Networks(GNNs) to image classification and the MoNET framework for dealing with geometric data in general. Their framework works by weighting the neighborhood aggregation through a learned scaling factor based on geometric distances. Velickovic et al. (Velickovic et al. 2018) proposed a model using self-attention for weighting the neighborhood aggregation in GNNs. Although this model was a sub-model of the MoNET framework, it provided extraordinary results on other datasets, namely Cora and Citeseer, two famous citation networks (Sen et al. 2008), and on the FAUST humans dataset (Bogo et al. 2014).

Although Shi and Malik's seminal paper applied graph-based methods directly to images by converting each pixel to a graph node for image segmentation, smaller graphs can be generated with lower-level representations. Each segmentation region can be the natural choice for nodes of a graph but generating accurate segmentation results is still an open problem. Superpixels might be the middle ground between pixel-based graphs and object-related region-based graphs. Superpixels group pixels similar in color near each other into meaningful representation units called segments (Stutz, Hermans, and Leibe 2018). Several computer vision tasks

Figure 10: Sample Annotations for four key transition classes. (Row 1) Construction. (Row 2) Destruction. (Row 3) Cultivation and (Row 4) Decultivation. (Satellite images courtesy Google Earth).

can be performed on these over-segmented images, including depth estimation, segmentation, and object localization as in (Achanta et al. 2012). The work mentioned above on using GNNs for images, alongside the work on adapting self-attention for GNNs and the works for generating superpixels of images, form the pillars on which we based our experiments.

SplineCNN (Fey et al. 2018), and Geo-GCN (Danel et al. 2020) are two other models which extend MoNET frameworks to weight neighborhood aggregation based on geometric information. SplineCNN leverages B-spline base properties in their neighborhood aggregation procedure, while Geo-GCN engineered a learned distance function to perform data augmentation using rotations and conformations. Semi-supervised augmentation for classification is another technique for using GNNs with image data as in (Jiang et al. 2019). The main difference between their method is that they extract a feature vector for each image with a convolutional network and then build a graph on which they used their model. Although their technique is useful for semi-supervised learning. We use the vanilla GAT based classifier for a graph representing an image directly that is not comparable.

B Implementation Details

All the graph neural networks are trained using PyTorch. Optimization method is Adam with an initial learning rate of $1e^{-3}$. The learning rate increases by 0.1 if validation loss

Table 4: Classification of proposals for graph generation from images

Graph Type	Proposals
Pixel-based Graph	(Defferrard, Bresson, and Vandergheynst 2016; Edwards and Xie 2016; Liu et al. 2020)
Superpixel-based Graph	(Liang et al. 2016; Knyazev et al. 2019; Fey and Lenssen 2019; Liu et al. 2020; Wan et al. 2019)
Object-based Graph	(Jain et al. 2016; Qi et al. 2018a; Zhou and Chi 2019; Zhang, Campbell, and Gould 2020; Jiang, Koppula, and Saxena 2013; Tompson et al. 2014)

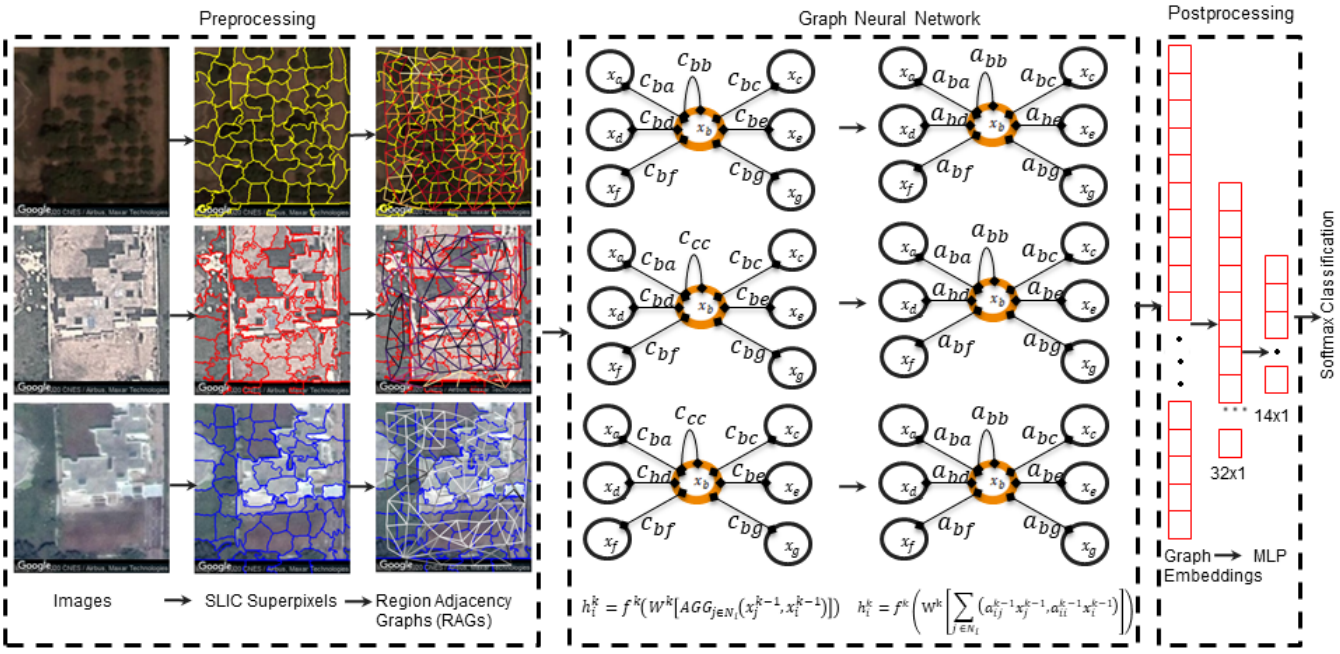


Figure 11: Flow diagram of our proposed spatial attention graph neural network (SAG-NN). Images are first converted into superpixels using SLIC, region adjacency graph is then constructed using these superpixels, finally spatial attention graph neural network is applied. Graph embedding are then used for classification via MLP. (Satellite images courtesy Google Earth).

does not decline for 20 epochs. Instead of using fixed number of epochs, we used early stopping criteria and patience for early stopping is 200. We have kept the same train, validation and test splits for all the datasets, i.e. 70%, 15%, and 15% respectively.

C Qualitative Analysis

Fig. 9 shows the sample annotations for *Construction* transition class. In Fig. 9 (Row 1), SAG-NN-E with voting mechanism classifies it as *Cultivation* which is clearly wrong as it can be seen from the middle and last image that the land has undergone the *Construction*. This type of misclassification is expected from the model since there are two transitions in three images of geolocation. The voting mechanism tends to get confused when multiple transitions are present in an example. But our proposed model 'STAG-NN-BA-GSP' correctly classifies it as *Construction*. In Fig. 9 (Row 2) our all models: SAG-NN-E, STAG-NN-BA-GCP, and STAG-NN-BA-GSP classify it as *Construction*. In Fig. 9 (Row 3), SAG-NN-E and STAG-NN-BA-GSP correctly classify it but STAG-NN-BA-GCP confused it with *Destruction* perhaps because in this example one building is removed while multiple others were added.