Satellite Image Classification Using CNN and GCN Models

Gabriel Treutle, 3rd year Data Science Major, treutleg@msu.edu
Hyperparameter Tuning | CV

Mehrshad Bagherebadian, 3rd year Computer Science Major, baghereb@msu.edu
Hyperparameter Tuning | CV | Graph-Based Learning

Aaryan Naruka, 3rd Year Computational Data Science Major, narukaaa@msu.edu
Model Training | Testing Data

Vish Challa 3rd Year Computational Data Science Major, challavi@msu.edu
Data Preprocessing| Augmentation Specialist

Ryan Krupp, 3rd year Computational Data Science Major, krupprya@msu.edu
GitHub Manager | Model Trainer

Pratham Pradhan, 3rd Year Computational Data Science Major, pradha11@msu.edu
Evaluation and Metrics Specialist

Abstract

Our idea is to classify satellite images based on what region, or biome, of earth that the image belongs to. This project will use the EuroSAT dataset, which is a collection of satellite images sorted into different land types, to create and compare machine learning models for identifying biomes. The dataset is about 94.7 MB which means it will be easy to work with, but it still includes a variety of images across ten land categories, representing different types of land cover such as forests, agricultural areas, urban regions, and water bodies. With all these diverse categories we will be able to explore the effectiveness of our machine learning models in classifying images based on various biomes.

We will use a Convolutional Neural Network (Resnet 18 specifically) as a baseline because they are well-known and widely used method for image recognition projects. CNNs are very effective at recognizing patterns and features in image data, making them an ideal strategy to be used for this project. In addition to CNNs, we will experiment with other models, such as Graph Neural Networks (GNN) or different types of neural networks, to evaluate their potential in improving our model's classification accuracy. GNNs can capture relationships between neighboring pixels or regions, which could enhance the understanding of spatial patterns in satellite imagery.

Our goal is to compare how well these models work, identify their strengths and weaknesses, and gain a deeper understanding of the machine learning techniques we are learning about in the context of our project. By doing this project, we want to contribute to the ongoing research in remote sensing and land cover classification. This project is inspired by past research that successfully applied CNNs to this dataset. We hope that we can add to that previous work and ultimately hope our findings can help support environmental monitoring, land management, and biome preservation. As the GitHub manager, Ryan Krupp will own the respective repositories and ensure seamless version control. The team will begin by applying our baseline model, the CNN, with ResNet as our architecture. Typically, this will entail traditional hyperparameter tuning and cross validation to assess generalization across land cover types. This will be worked on by Gabriel and Mehrshad. Training and testing the model will be accomplished by Naruka with the rest of the team contributing if need be. Pratham will determine the appropriate metric(s) to evaluate the CNN. Consequently, the GNN will be implemented with similar role

distributions and metrics. However, the team aims to test different versions of this neural network such as GCNs (Graph Convolutional Networks). As the classification task is purely based on image pixel features, GCNs may outperform the traditional CNNs.

Introduction

Knowing what kinds of land cover and biomes are in an area is important for things like keeping track of climate change, helping farmers plant their crops, and architects develop cities. Satellite images are helpful because they let us see large areas of land from above. We can use machine learning technologies to classify different biomes and separate them from these satellite views of the world. This is important for environmental protection and crop preservation.

This project will use the EuroSAT dataset, which is a set of satellite images sorted into different types of land. The dataset is small (about 94.7 MB), so it's easy to work with, but it still has a good variety of land types like forests, farms, cities, and water. This gives us a chance to see how well machine learning can tell these different areas apart.

The goal of our project is to see which models work best and learn more about using machine learning in the context of satellite images. We want to add onto previous works based on similar ideas using the satellite data set and classifications. What we learn could help with things like protecting the environment, managing land, and other projects that need accurate land cover information from satellite images. That is why this project is important to us and others in the world.

Dataset/Methodology

2.1 Dataset & Preprocessing

The EuroSAT dataset consists of 27,000 RGB satellite images, categorized into ten land cover classes, including forests, industrial areas, water bodies, and annual crops [**Figure 1**]. Each image is 64×64 pixels, derived from Sentinel-2 satellite data. The dataset provides a balanced class distribution, making it well-suited for benchmarking deep learning models in remote sensing.

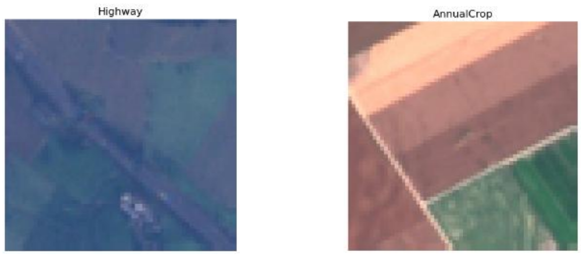

Figure 1: Satellite image of two target classes: a highway and an agricultural land.

Fortunately, EuroSAT has been normalized and preprocessed after data collection. This allows the team to dive into model selection and training.

All images were resized to 64x64 pixels to ensure consistency and optimize training time. In the beginning, data augmentation techniques were implemented to promote model robustness and avoid overfitting. These encompass random flips — vertical and horizontal — to increase the variety of the training samples. Additionally, each image was standardized using the mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225]. These values align with the standards of ImageNet, showcasing compatibility with the pre-trained model weights.

2.2 Dataset Splitting

As standard practice, the ResNet-18 dataset was split into training, validation, and testing segments to train and test the model. 70% of the set was used to solely train the model, while 10% was used exclusively for validation during the model's training. The rest of the dataset (20%) was used for testing.

Due to the unique architecture of the GCN, the nodes that represent each image are also split into training, validation, and testing split of 70%, 10%, and 20% respectively. It is important to note, however, that due to the location of the nodes in the adjacency matrix they are not split into separate data sets. Each node inside of the adjacency graph structure is labeled with either train, validation, or test for training. If the structure were split into training, validation, and testing, the nodes would lose their relationship.

2.3 Model Architectures

As for the model architecture itself, the baseline of the ResNet-18 model was utilized, pre-trained on a large ImageNet dataset. As a reminder, this was selected due to the track record of generalizing well to new image classification tasks. Keep in mind that the original fully connected (FC) layer was modified to align with the output predictions of the ten EuroSAT land cover categories. Originally, the layer included the 1000-class output that was then tweaked to address the 10-class prediction problem.

For the more in-depth model, we opted to use the ResNet-18 model we had as the baseline for the base of a GCN model. This way, we could leverage the pretrained ResNet-18 model for the GCN architecture. This entails passing the images through the ResNet-18 model to begin with, but then instead of outputting its predictions, it ends before the final layer and gives a feature vector instead. This feature vector is then passed into a function to build an adjacency graph, where the most similar images (or what are now feature vectors) are grouped together through a K nearest neighbor algorithm. The K nearest neighbor algorithm used cosine similarity to decide groupings. The GCN is then able to take both the feature vectors from the ResNet-18 model, and the adjacency matrix created through the K nearest neighbor algorithm to form its own outputs. This use of the adjacency matrix essentially allows the GCN to account for how similar images are being classified by using any given feature vectors neighbors, which allows enhanced information during decision making.

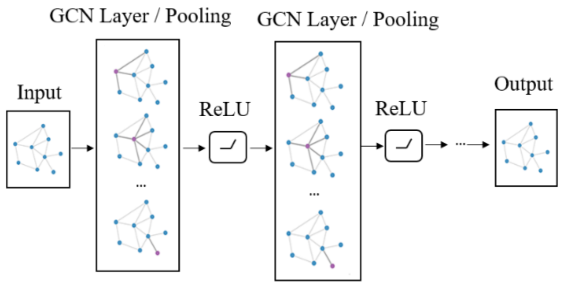

Figure 2: GCN Architecture

2.4 Training Strategy

For both the Resnet-18 model and the GCN implementation the training process was performed over twenty epochs using mini-batch gradient descent, with each mini-batch containing 64 images. 64 images were chosen to strike a balance between training time and model performance. In standard practice, cross-entropy loss was implemented for the multi-class classification tasks. As for optimization, the popular adaptive optimizer AdamW was employed with weight decay. AdamW is known for its ability to achieve rapid convergence without overfitting. The initial learning rate was set at 0.001 but then changed to 0.0001 as it increased final convergence in both models. For both we made sure to set the model to. eval() and torch.no_grad() during validation and testing so that new gradients would not be calculated and dropout (which we did not end up using) and batch normalization would not take effect.

For the ResNet-18 model, it took about 20 epochs before overfitting began to take effect, with about the same amount for the GCN model.

Both models performed better when data augmentation such as horizontal and vertical flips were used. The models converged more slowly but saw increased accuracy. As the name suggests, the ResNet-18 model is 18 layers deep, which contributed to a longer run time on our local machines. When the number of layers in the GCN that were being used on top of the ResNet18 model was two, the best balance was struck between run-time and accuracy. With more layers training the model with respect to time became unfeasible, but with two convolutional layers in the GCN wrapper the model was able to train in a reasonable amount of time with accuracy around 91%.

Increasing the number of neighbors for the GCN past 5 did not seem to contribute meaningful results to the model at any given point. This was tested with neighbors of count to 20, and as low as 1, with 5 bringing the best results. 1 is the natural lower limit, as if it was set to 0 there would be no relations being found between neighbors at all.

Adding dropout was entirely unsuccessful for both models. It might be because the dataset is too small, but the addition of dropout in any magnitude only decreased the accuracy of the models. This decrease was mitigated if the models were run for more epochs but never contributed positively to the models' success. In the case of a dropout rate of 0.3 for the ResNet18 and GCN respectively, the models that achieved the highest accuracy of 96.7% and 91.22% for the ResNet18 and GCN models respectively dropped by 3% and 6% in accuracy each. One could assume that this might be because the dataset is too small to handle dropout, or possibly since Resnet-18 employs batch normalization which may have already accounted for the variance.

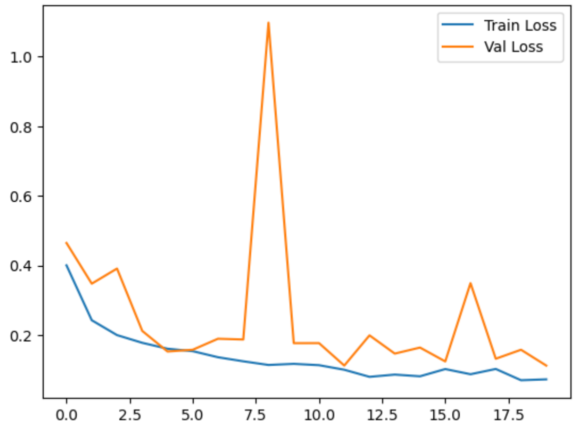

Figure 3: Multi-class confusion matrix for Resnet-18

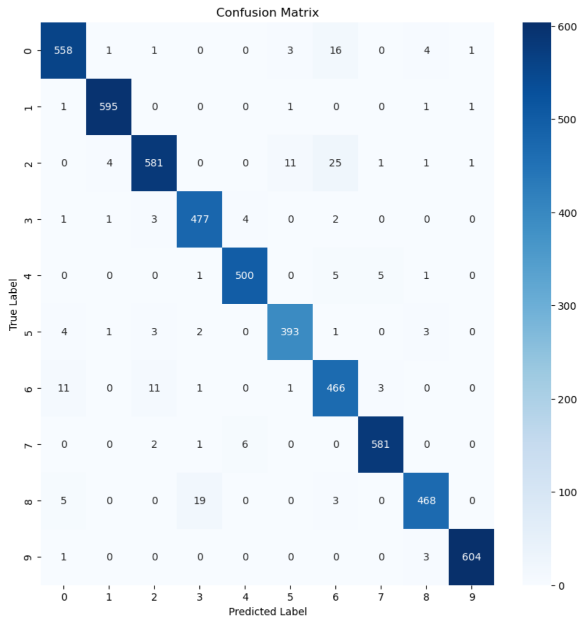

Figure 4: Training vs Validation Loss Over Twenty Epochs ResNet-18

2.5 Evaluation Metrics

To assess model performance, the following metrics were used: Accuracy, f1, precision, and recall.

Results from the confusion matrix [**Figure 3**] and loss curves [**Figure 4**] provide insights into classification errors and model stability during training for the ResNet18 baseline model. The confusion matrix provides a visual representation of the model's performance across all ten classes, emphasizing areas where the model excelled as well as struggled to a small extent. The accuracy indicates the sum of the true positive and the true negatives divided by the total number of predictions. The model achieved a final testing accuracy of 96.72%, precision of 96.7%, recall of 96.7%, and f1 of 96.7%. These results are strong, but not necessarily as high as was achieved in other tests. Annual crops and forests were large contributors to the errors, frequently being confused about the permanent crop classification. Conversely, permanent crops were often confused for annual crops or herbaceous vegetation. It is possible this is due to similarities in vegetation between these classes.

The GCN model achieved lower metrics than that of the ResNet-18 model. See [**Figure 5**] and [**Figure 6**] for results on the model. While the GCN model might appear not to have bottomed out yet in terms of loss, further testing had shown previously that past 20 epochs the change was minimal to none in test metrics. The GCN model had a final testing accuracy of 91.22%, precision of 91.4%, recall of 91.22%, and f1 of 91.2%. There was similar struggle points for the GCN compared to the baseline ResNet-18 model, but they were typically more exacerbated. For example, where the ResNet-18 model only misclassified 11 herbaceous vegetation for permanent crop, the GCN misclassified 64. Another struggle point was again in misclassifying “Annual Crop” (0) and “Herbaceous Vegetation” (2) for the class “Permanent Crop” (6). The classes with the highest general accuracy for both models were “Sea Lake” (1), “Forest” (2), and “Herbaceous Vegetation” (3). The most accurate class for prediction with the ResNet-18 model was the “Sea Lake” (9), while for the GCN it was the “Herbaceous Vegetation” (2).

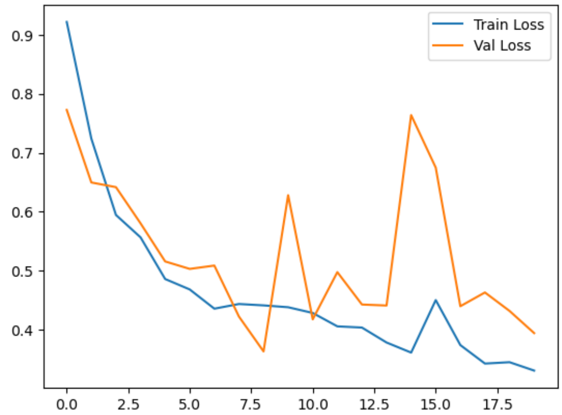

Figure 5: Training vs Validation loss for GCN

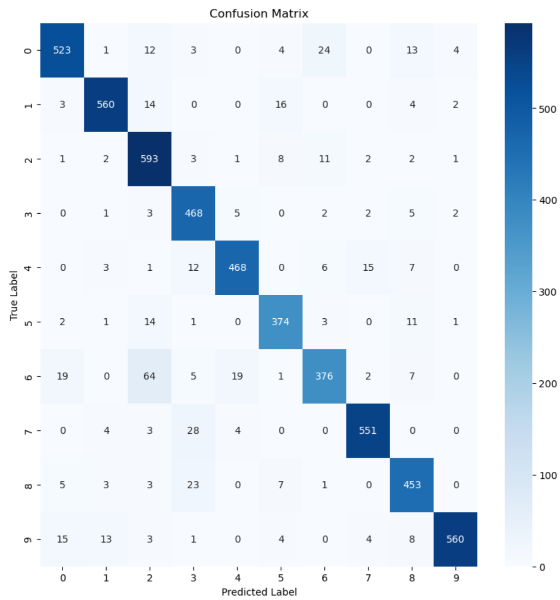

Figure 6: Multi-class confusion matrix for GCN

3. Literature Comparison

We are basing our research on a couple of different papers. The first is “*Introducing Eurosat: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification*”. In the paper, they analyzed the effectiveness of the BoVW, GoogleNet, and ResNet50 models on the data, pretraining them on the ILSVRC-2012 dataset. The papers also explored the most effectiveness of different spectral bands to use for image classification and found the RGB band to be the most effective. In general, research has found that deep CNNs tend to do well for land classification. The neural networks used previously for this research have all been either trained from scratch or supplemented by a pretrained network trained on similar images.

The implementation that we went through for our baseline CNN model was the ResNet18 pretrained model, and it was used to classify images from the EuroSat dataset. The model that was constructed and created through extensive testing resulted in a model with a test accuracy score of 96.58%. This was tested over 10 epochs, and this model illustrated that it worked and was efficient because, as stated by Zhu et al. (2017), “transfer learning from models pretrained on ImageNet… significantly boosts performance.” This was significant because our model achieved high test accuracy by doing the same thing as a principal method in a research paper. This is also important to our model's performance because in the model we can see that there is strong performance in our model. After all, the researchers followed the same ideas as we used, even though our model was constructed over a smaller number of training epochs. When looking at the other values in our model, for example, the validation loss and the confusion matrix, which demonstrate the proper generalization of the model. This is essential for the effectiveness of the model and was illustrated by the low validation loss and clean confusion matrix. Additionally, when compared to the research of Penatti et al. (2015), it was said that "features learned from everyday object images can be successfully transferred to remote sensing applications." This is very important because even though the model was initially used for ImageNet classification, it is now able to identify and classify satellite-based images.

After we created a reliable CNN we moved onto a Graph Convolutional Network (GCN) that came in the form of a wrapper on top of the CNN. To give a general overview of how a GCN operates, it consists of nodes and edges that determine the path and connections for the graphs. This is the difference compared to CNN, however, GCN works well with graph-structured data and feature-based relationships. In the implementation of our GCN, we again classified images based on the EuroSat dataset, and this model achieved a test accuracy score of 91%. This illustrates that the model was efficient across all the classes of data. However, when comparing the accuracy score, it is obvious that the accuracy score of the GCN is lower than that of the CNN. This could have happened because the EuroSat dataset had more independent images, and this led to a decrease in strong connections.

Now, looking at the test accuracy score, we can understand our model’s effectiveness, and the research will demonstrate the true effectiveness and efficiency of the model. Additionally, as Li et al. (2020) reported, using GCNs for scene understanding helped reduce confusion between visually similar categories. This can be confirmed with the numbers of the test accuracy and the F1 score, which is 91.2%. Additionally, when we look at other research by Chen et al. (2021), applying GCNs to hyperspectral image classification improved accuracy by modeling spatial and spectral relationships between data points. I believe this was mostly true as there was very good value that was achieved because of the use of the GCN, but it did not achieve higher value since the limitations within the EuroSat dataset. In the research work on the GCN model, they achieved an overall accuracy of 93.6%, which was like our model.

Finally, looking at our results and comparing them with the values or models of the research, we can conclude that our CNN model worked very well in classifying satellite images. The GCN was also effective and worked well with what it had done; however, if there were no limitations in the dataset of independent images, the GCN would have been a more powerful tool. However, both models followed information that was supported by the research paper, which made them effective models.

4. Conclusion

In this project, we explored the application of deep learning techniques for the classification of satellite images using the EuroSAT dataset. Our primary goal was comparing the effectiveness of traditional Convolutional Neural Networks, specifically the ResNet-18 model we implemented, with a newer and more recent model built on the existing knowledge of CNN's, the Graph Convolutional Network (GCN). The GCN architectures are adapted to image data through graph representations, which is suitable for a multi-class classification task using the EuroSAT dataset.

Through careful experimentation, our group was able to demonstrate that the ResNet-18, pre-trained on ImageNet and fine-tuned for the 10-class classification task, provided as a strong baseline model with high accuracy and constant generalization.

Thus, we extended our exploration to GCNs by creating a graph-based wrapper around the ResNet-18. This way, instead of only relying on independent image predictions, we were able to encode similarity relationships between the images using a k-nearest neighbors graph (KNN) where nodes represent the image embeddings, and the edges reflect cosine similarity in the feature space. Using these graph structures with the GCN, we aimed to enhance classification by using relational context across samples. This transition in architecture allowed our group to see the potential of the GCNs to generalize spatial patterns beyond the CNNs capabilities, providing us with a new perspective on structured learning for satellite images and classification tasks.

The results from our experiments demonstrated a clear performance gap between the two models. The ResNet-18 CNN achieved better results across all evaluation metrics, including accuracy, precision, recall, and F1-score, with a final testing accuracy exceeding 96%. In contrast, the GCN model, although achieving a great performance of 91% accuracy, showed greater variability across the classes and a higher chance of misclassification especially among similar land types such as agricultural fields and herbaceous vegetation. This discrepancy shows that while the GCN can offer a useful way to connect similar images, their performance depends a lot on how the graph is built. If the images don't have strong connections to each other, or if the graph doesn't capture enough overall structure, the model may not learn as effectively. These insights not only highlight the strengths and limitations of each architecture but also point toward opportunities for further improvements in hybrid models.

5. Future Works

With a very high testing and validation accuracy, the CNN Resnet-18 model demonstrated strong predictive performance. There exist minor fluctuations in validation loss and other miscellaneous areas that can be further fine-tuned and experimented to maximize accuracy and other evaluation metrics. As stated before, the next step is to utilize the Graph Neural Networks (GNNs), particularly the Graph Convolutional Networks (GCNs). We hope that GCNs will leverage spatial relationships and satellite-image context, potentially capturing more minute relationships unlike CNNs.

However, there is room for improvement. With a loss of over 5% in accuracy as well as a decrease in other metrics, there are directions for future work. The latest architecture we used for classification faced several challenges that limit its accuracy compared to the pure CNN baseline. A primary problem may be the dependency on the batch-wise k-NN graphs. They do not capture the global relationships across the set. This promotes inconsistent patterns, while avoiding spatial context. An example of this would be the geographic coordinates of image patches being ignored. Another limitation is the lack of depth of the GCN architecture. With two layers, the network may not capture other complex relationships and restrict hierarchical feature learning.

One way to address these limitations is by incorporating other architectures with the strengths of the baseline and GCN models. First, a global spatial spatial-semantic graph can be built (Zhou). This is accomplished by computing the ResNet-18 features for the training samples prior to merging semantic similarity and geographic proximity. This should preserve relationships between adjacent and semantic similar regions. This could address the global relationship issue.

As for the architecture, GCN can be improved through graph attention networks with the addition of residual connections (Veličković). This allows the architecture to mitigate the gradient vanishing problem.

With the help of the global graph as well as GAT with residual connections, accuracy is expected to increase along with other metrics. The combination of both methods enables cross-batch relational reasoning and promotes stabilized feature propagation. Furthermore, different stages of training and experimenting with additional data augmentation may address any class imbalances. However, it’s crucial to preserve the original data, so something as simple as SMOTE may address any imbalances if need be.

Overall, with these potential additions, the bridge between CNN baselines and more complex networks like GCNs may provide a layout for leading, interpretable satellite image analysis and deploy a basis for applications in environmental planning and urban monitoring.

References

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114-133. https://doi.org/10.1145/322234.32224.

Association for Computing Machinery. 1983. *Computing Reviews*, 24(11):503-512

Chen, Yushi, et al. “Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks.” IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 10, Oct. 2016, pp. 6232–6251, elib.dlr.de/106352/2/CNN.pdf, https://doi.org/10.1109/tgrs.2016.2584107. Accessed 8 May 2021.

Cheng, Gong, et al. “Remote Sensing Image Scene Classification: Benchmark and State of the Art.” *arXiv.Org*, 1 Mar. 2017, arxiv.org/abs/1703.00121.

Huang, Liwei , et al. “IEEE Xplore Full-Text PDF”: *Ieee.org*, 2025, ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10330561. Accessed 13 Apr. 2025.

Jesse Dodge, Zhuyun Hu, Jonathan Herzig, Gabriel Ilharco, Sahar Sadeghi, Kanishka Misra Goyal, Danqi Chen, Roy Schwartz, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2023. The BLiMP Supplement: Investigating the Influence of Training Data on the Evaluation of Language Models. *Zenodo*. https://doi.org/10.5281/zenodo.7711810.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). Graph Attention Networks. *ArXiv*. https://arxiv.org/abs/1710.10903. Accessed 12 Apr. 2025.

Zhu, Xiao Xiang, et al. “Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources.” *eLib*, IEEE - Institute of Electrical and Electronics Engineers, 1 Dec. 2017, elib.dlr.de/118694/.

Zhou J, Qin X, Yu K, Jia Z, Du Y. STSGAN: Spatial-Temporal Global Semantic Graph Attention Convolution Networks for Urban Flow Prediction. *ISPRS International Journal of Geo-Information*. 2022; 11(7):381. https://doi.org/10.3390/ijgi11070381. Accessed 12 Apr. 2025.