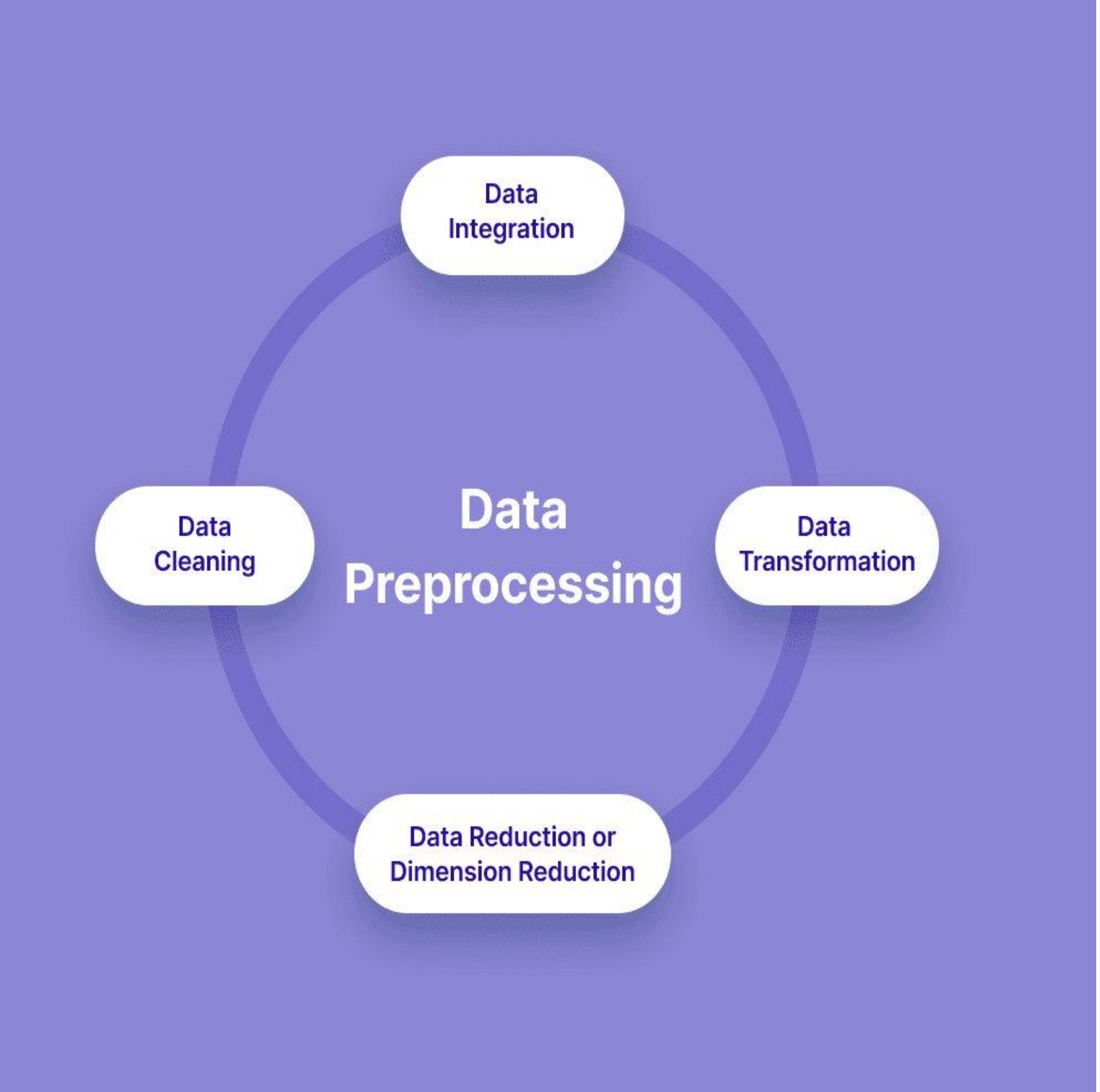
CNN and GCN for Satellite Image Classification

By: Ryan Krupp, Aaryan Naruka, Vish Challa, Gabriel Treutle, Mehrshad Bagherebadian

Abstract

- Our team goal for this project was to create an accurate image classifier for satellite image data.

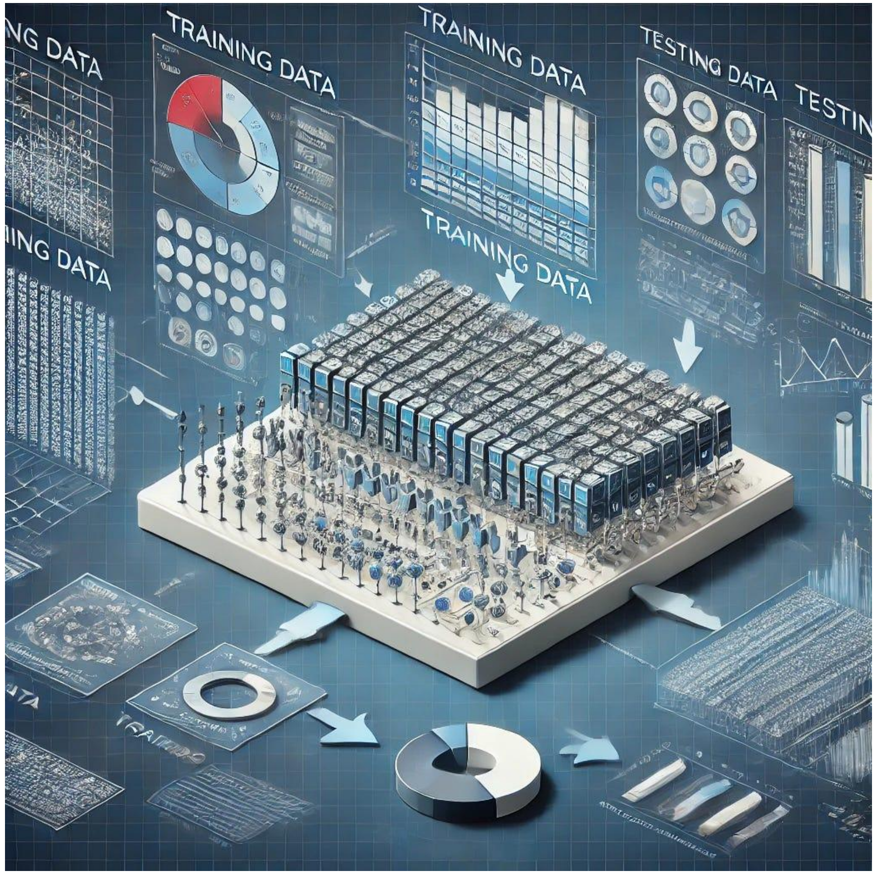
- To achieve this goal our team developed two models to classify satellite image data:
  - Our baseline model is a CNN (Convolutional Neural Network) model
  - Our more complex model is a GCN (Graph Convolutional Network) model



Dataset / Preprocessing

- Our dataset contains 27,000 RGB satellite images.
- All images are categorized into 10 land cover classes (e.g., forests, industrial areas, water bodies, crops).
- Each image is 64×64 pixels, sourced from Sentinel-2 satellite data
- Random horizontal and vertical flips to increase sample variety within the training data
- Standardization used ImageNet norms:
  - Mean: [0.485, 0.456, 0.406]
  - Std Dev: [0.229, 0.224, 0.225]
- Standardization ensures compatibility with pre-trained models like ResNet

Dataset Splitting



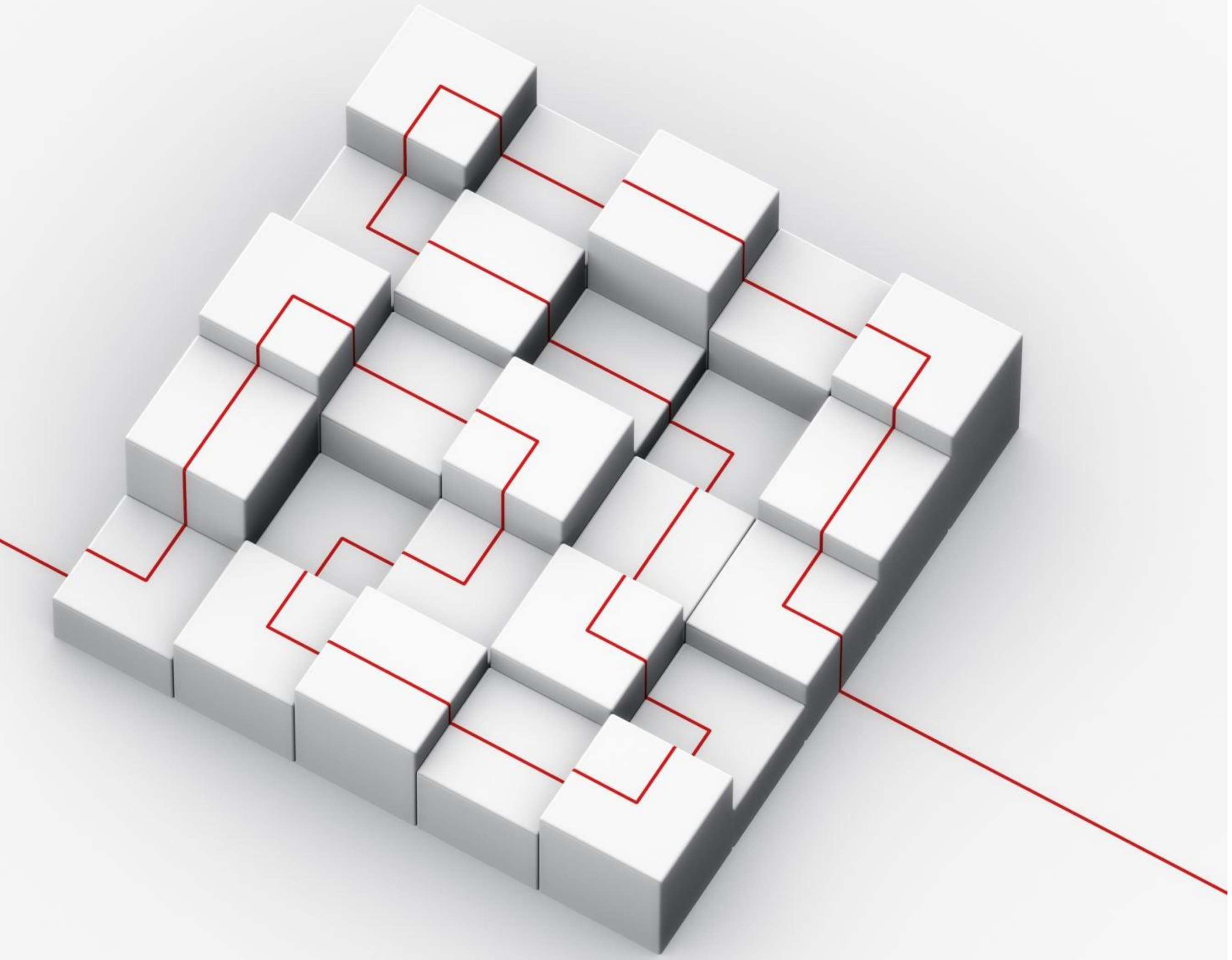
Dataset was split into Training, Validation, and Testing split of 70%, 10%, and 20% respectively

Due to architecture of GCN, the nodes that represent each image are also split into Training (70%), Validation (10%), and Testing (20%)

Nodes are not physically separated in the adjacency matrix

All nodes remain part of a single graph structure to preserve relationships

Labels for nodes are stored internally and used according to the split

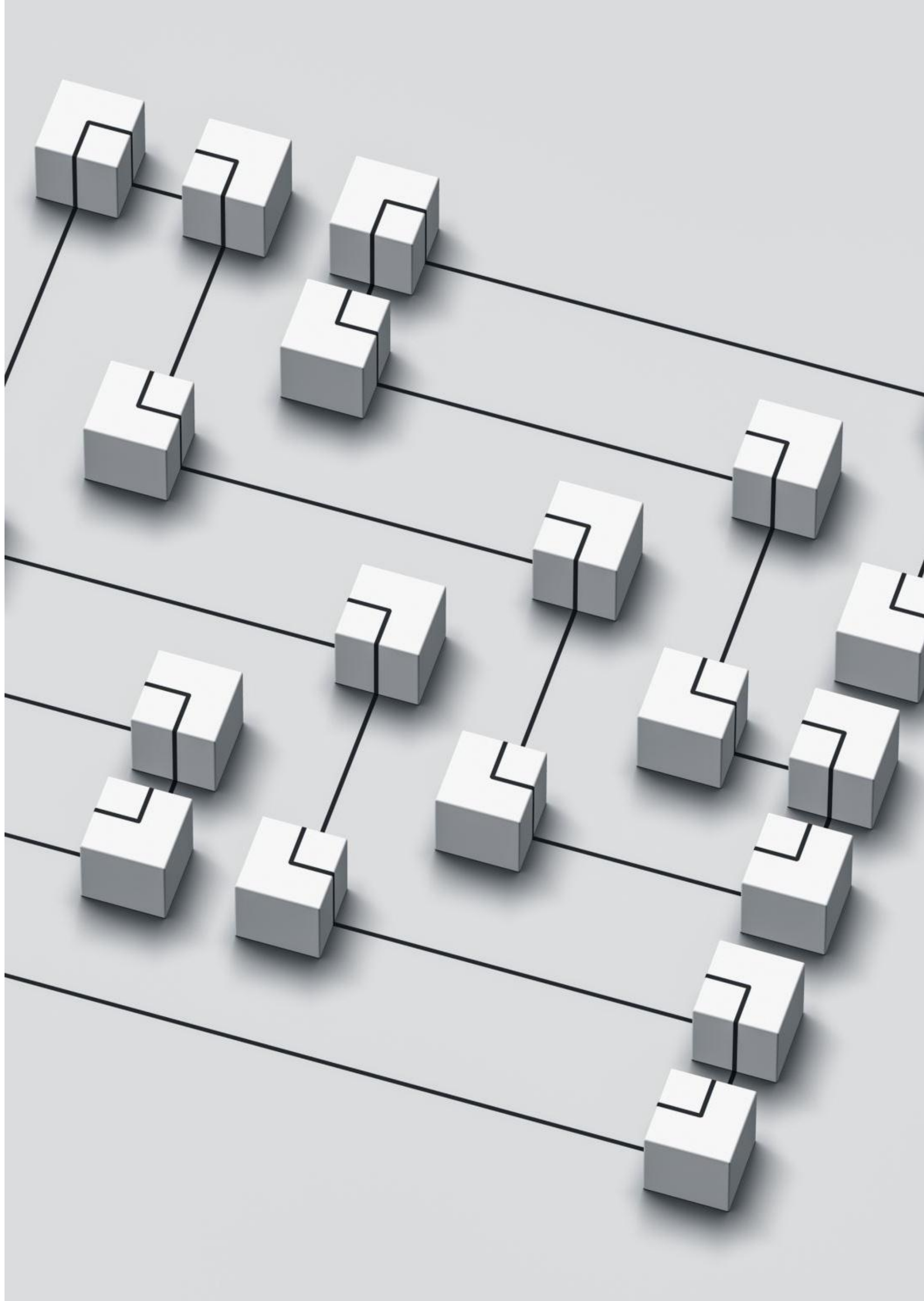


Model Architecture: Baseline Model

- Pretrained on ImageNet for strong generalization in image classification tasks.
- Original 1000-class output layer replaced with a 10-class output layer for EuroSAT land categories.
- Chosen for its strong performance on image-based datasets and pattern recognition.
- ResNet-50 Used in other projects, but run-time was unfeasible for testing our model

Model Architecture: In-Depth Model

- ResNet-18 used as a feature extractor by removing the final FC layer.
- Outputs a high-dimensional feature vector for each image.
- These feature vectors serve as node features in the GCN graph.



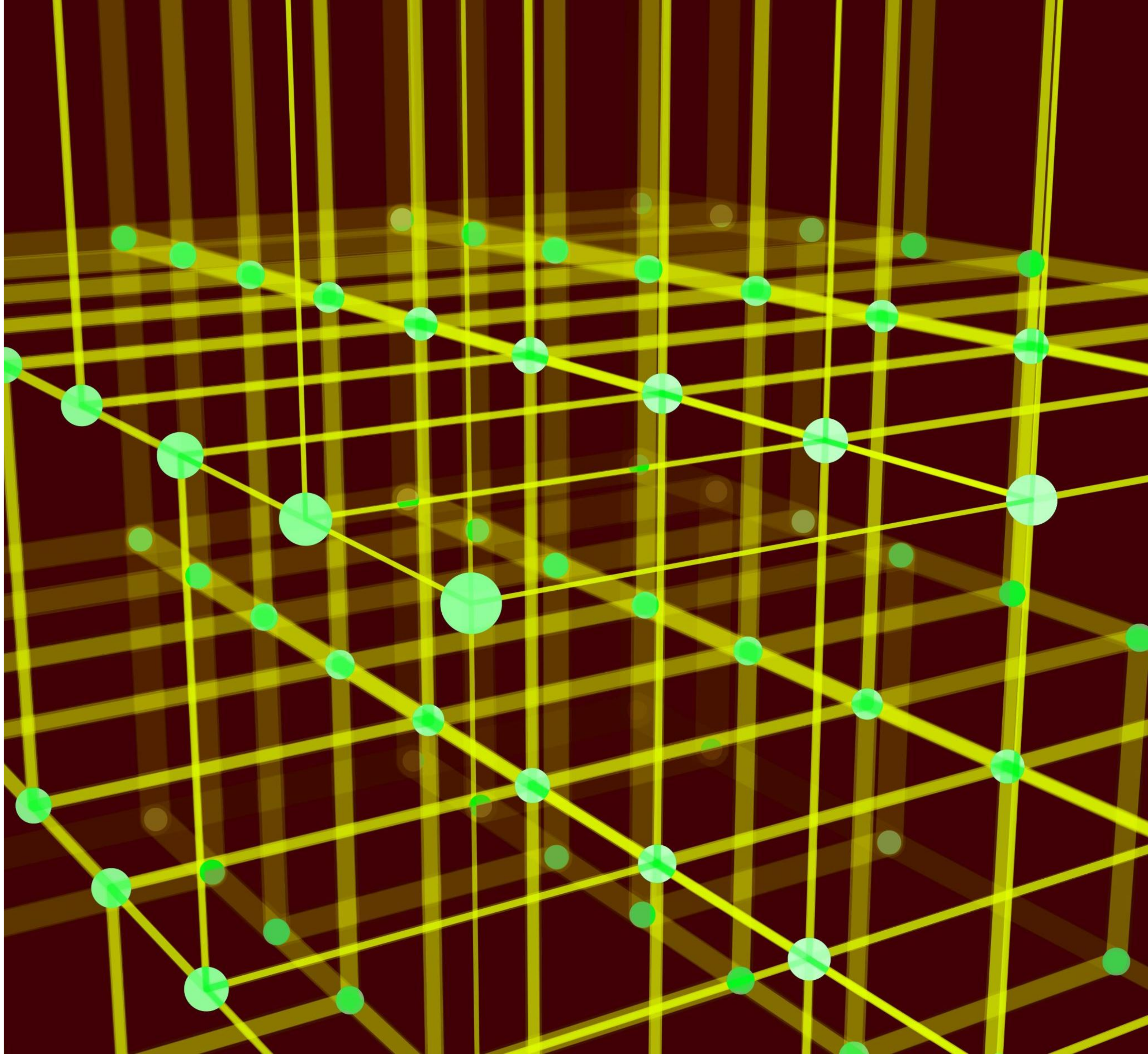
Model Architecture: In-Depth Model Cont.

•Built a graph using K-Nearest Neighbor (KNN) algorithm.

•Similarity calculated using cosine similarity between feature vectors.

•Nodes = images

• Edges = connections to most similar images.



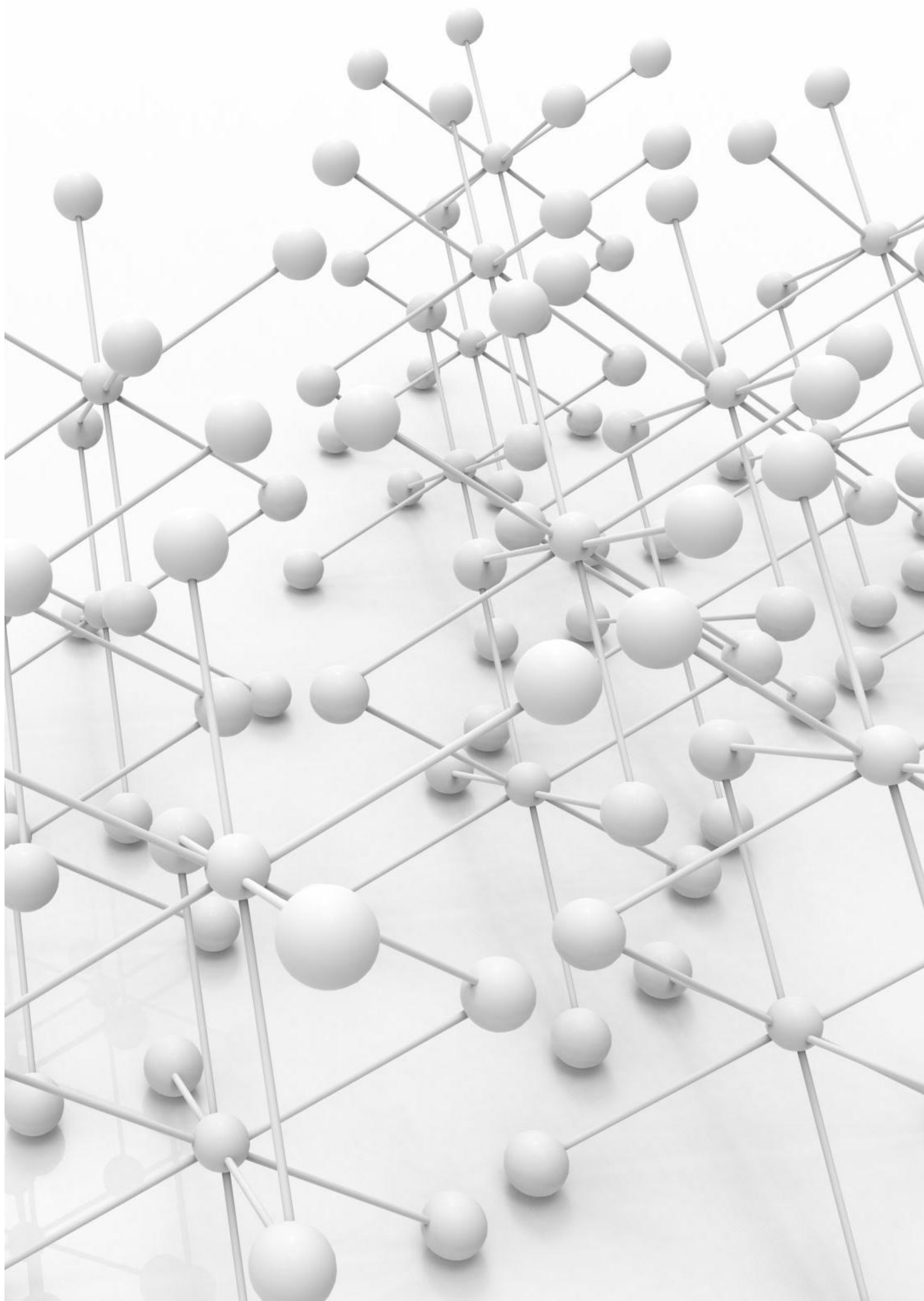
Model Architecture: In-Depth Model Cont.

•Node features from ResNet-18.

•Adjacency matrix from KNN graph.

•GCN is able to see “similar” images which impacts its predictions

•Allows for richer decision-making through structural relationships.



Training Setup

- Used 20 epochs for both architectures
- Used mini-batch gradient descent with a batch size of 64
- Loss Function: Cross-Entropy(Standard for these tasks)
- Optimizer: AdamW with weight decay
- Learning rate:
  - Started at 0.001
  - Lowered to 0.0001 to improve convergence
- .eval() and torch.no_grad() used during evaluation and testing

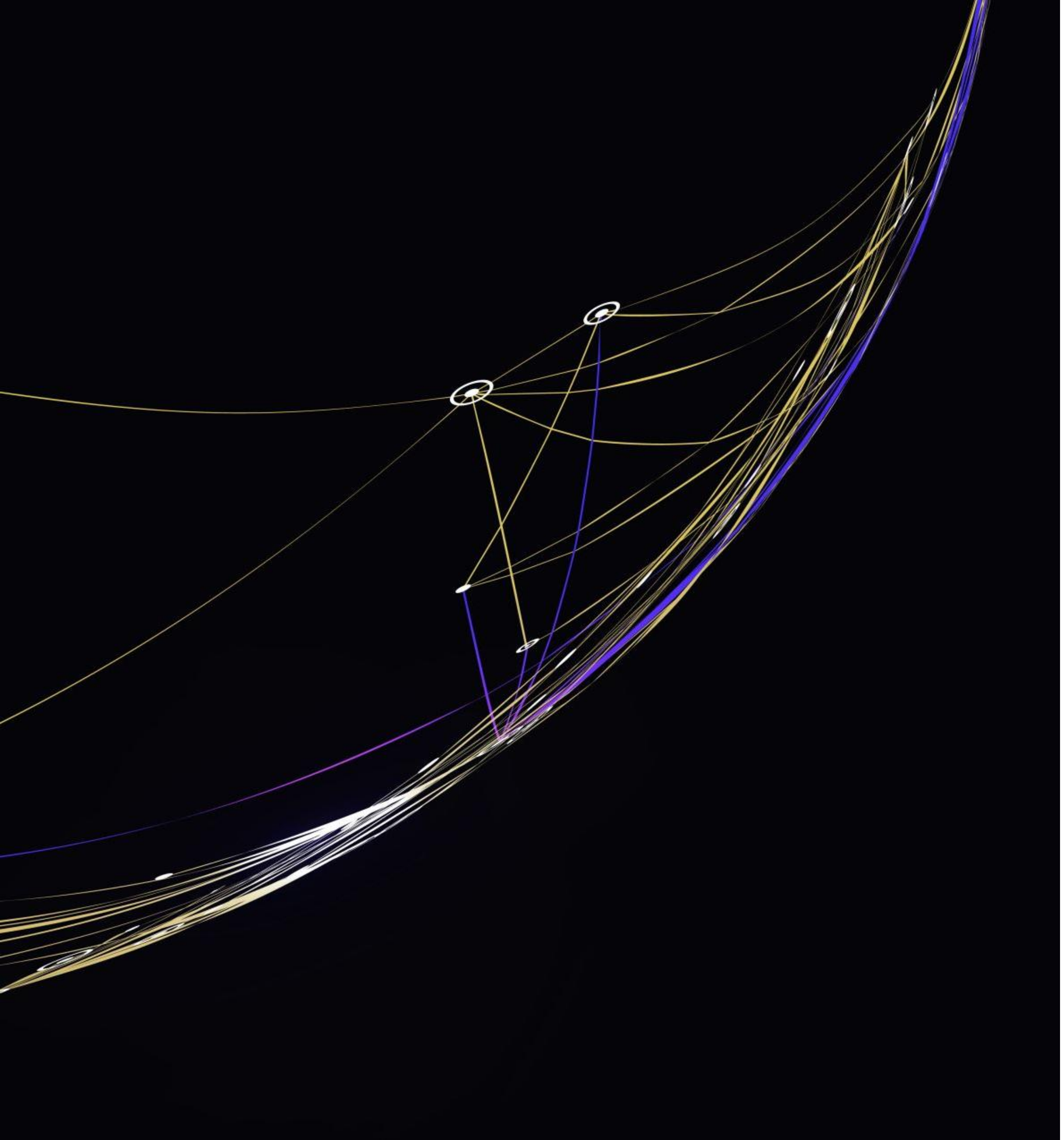
Model Observations

Both showed signs of overfitting past 20 epochs

Early signs in validation curve flattening

Dip in generalization past this point

- GCN with 2 layers: Best balance between run time and performance
- GCN with 3 layers: Training time became longer with decrease in metrics



Data Augmentation Effect

- Horizontal & vertical flips improved performance
- Slower convergence but higher final accuracy
- Helped mitigate overfitting in both models
- Models real life data better, as satellites are not always getting data from the exact same angle due to their travel

Number of Neighbors

•KNN tested with neighbor counts: 1 to 20

•k = 5 yielded the best results before no change

•k = 1: sparse graph, limited learning

•k > 5: no significant performance gain, increased computation time

Dropout Usage

- Dropout was not effective in this project
- ResNet18 with dropout had a 3% drop in accuracy
- GCN with dropout had a 6% drop in accuracy
- Dropout slowed learning, required more epochs, but never improved results
- Hypotheses:
  - Dataset too small for dropout to be effective
  - ResNet18 already uses batch normalization, possibly reducing the need for dropout

Evaluation Metrics

Precision

Accuracy

Recall

F1-Score

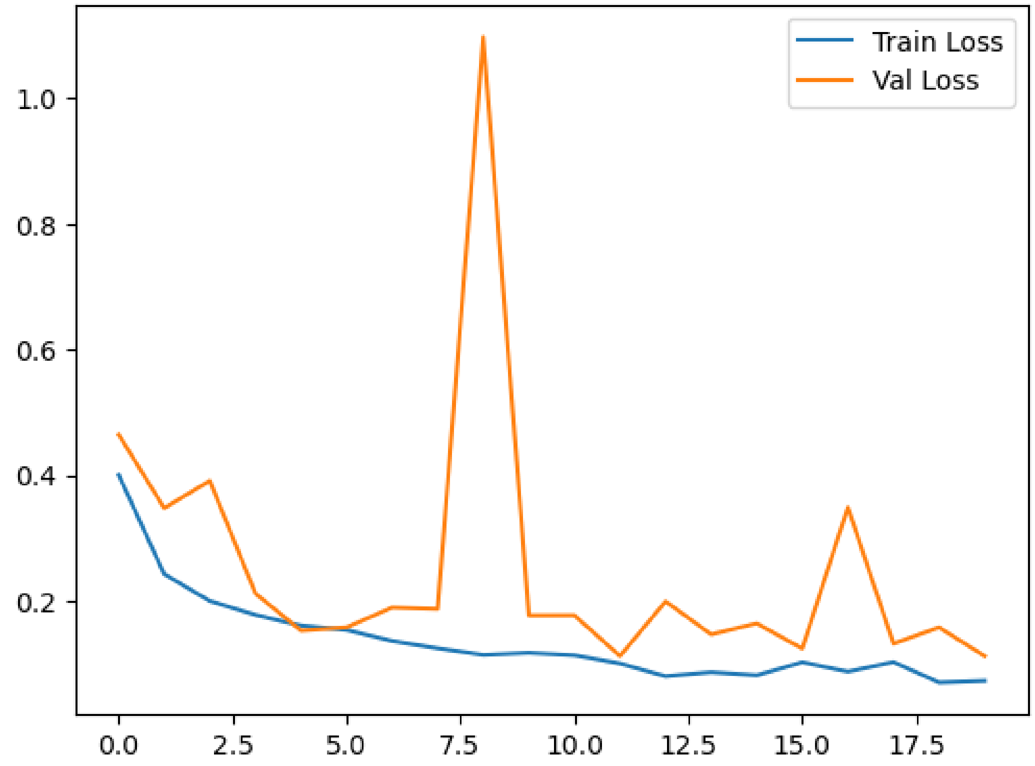
ResNet-18 Baseline Results

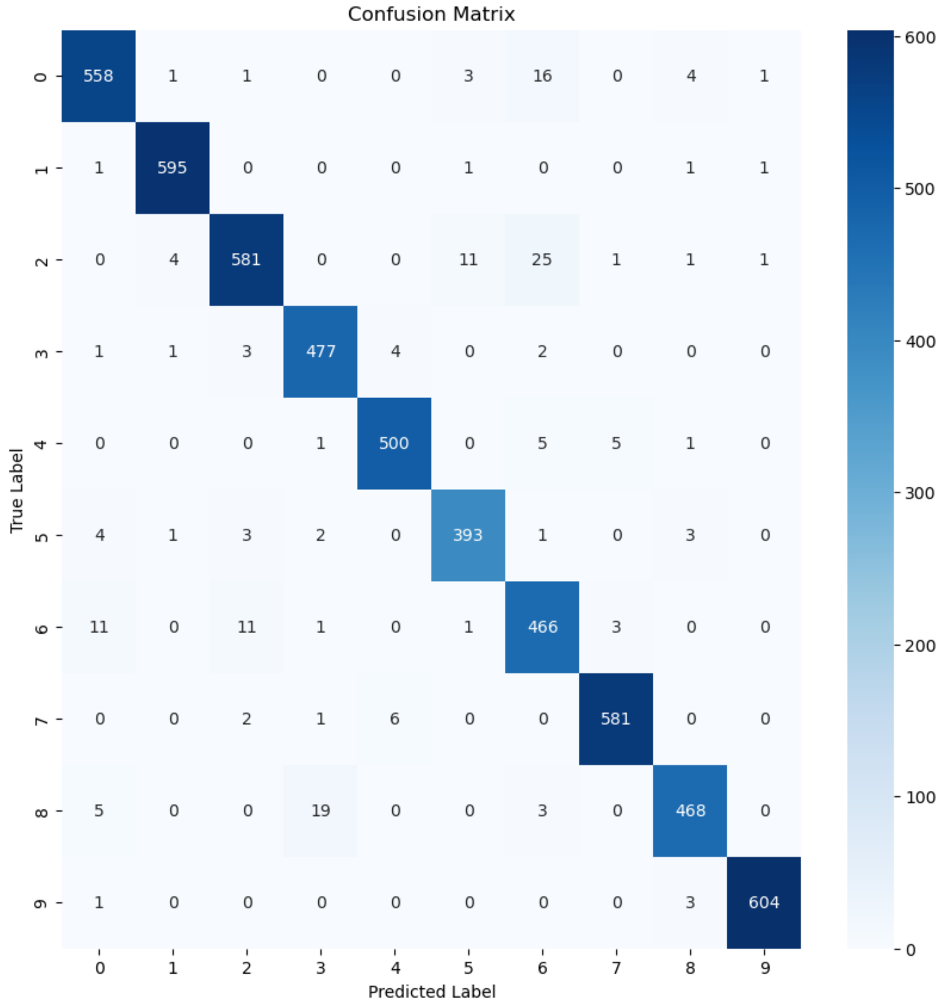
•Accuracy: 96.72%
•Precision: 96.7%
•Recall: 96.7%
•F1 Score: 96.7%

Strongest class predictions:

- Sea Lake (9), Forest (1), Herbaceous Vegetation (2)

ResNet-18 Visuals



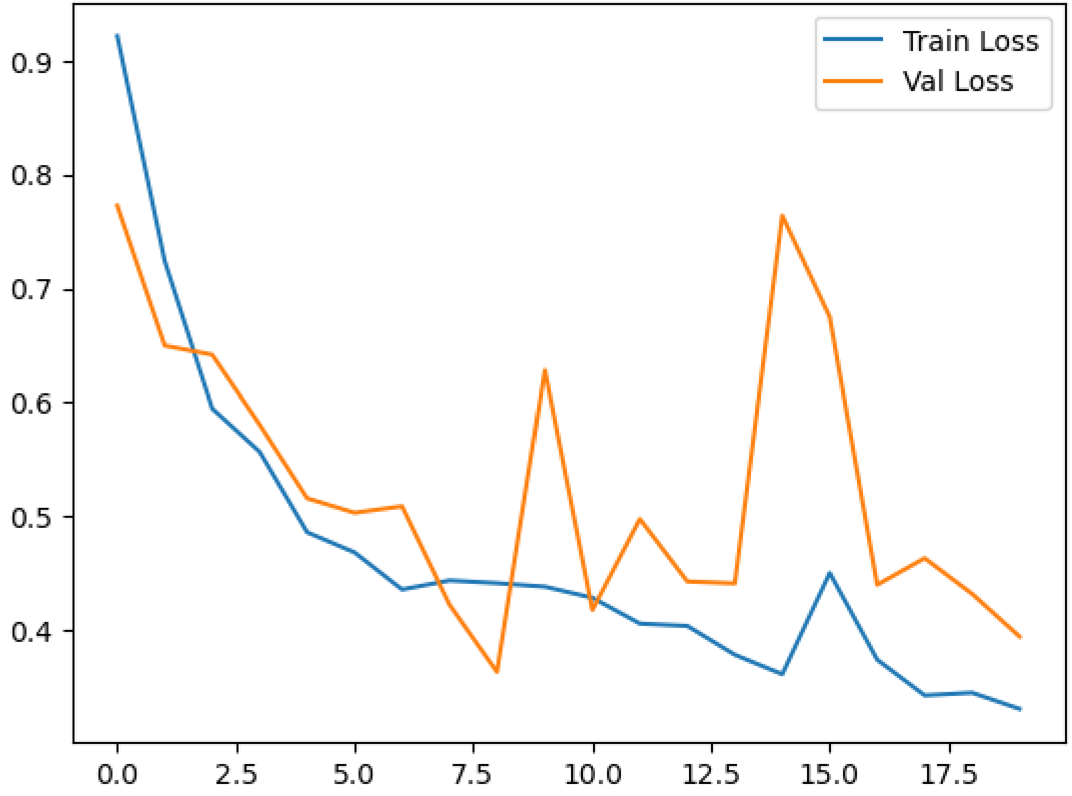


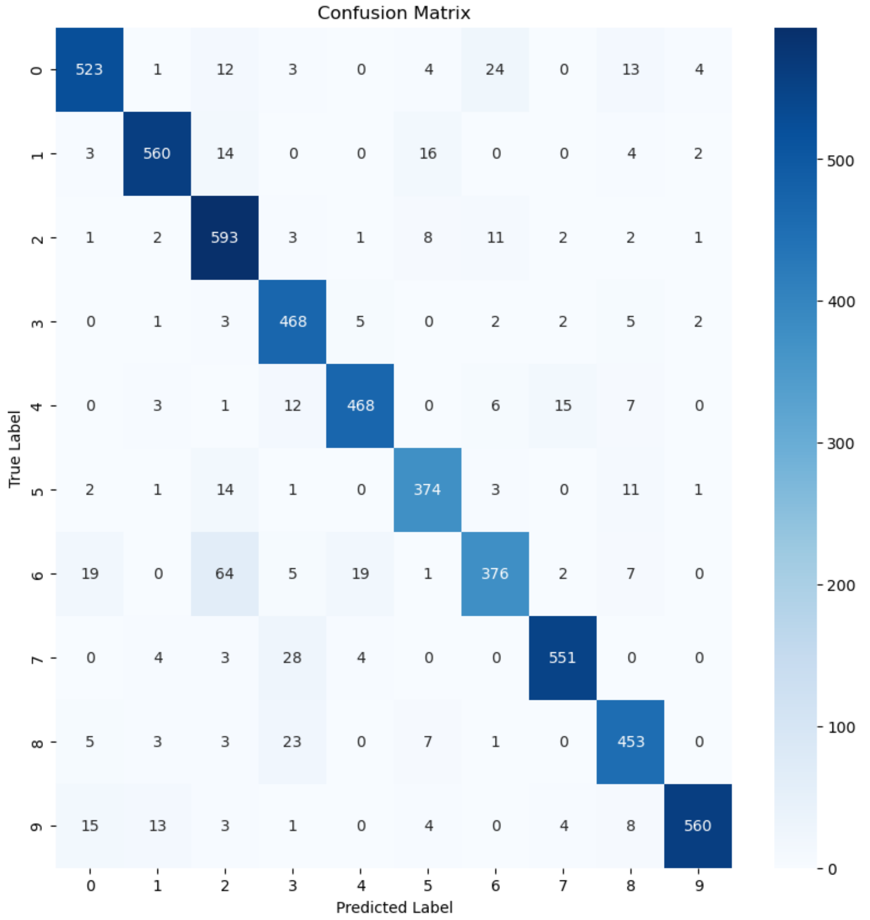
GCN Results

Final Testing Metrics:

- Accuracy: 91.22%
- Precision: 91.4%
- Recall: 91.22%
- F1 Score: 91.2%
- Would overpredict Herbaceous Vegetation(2)

GCN Visuals

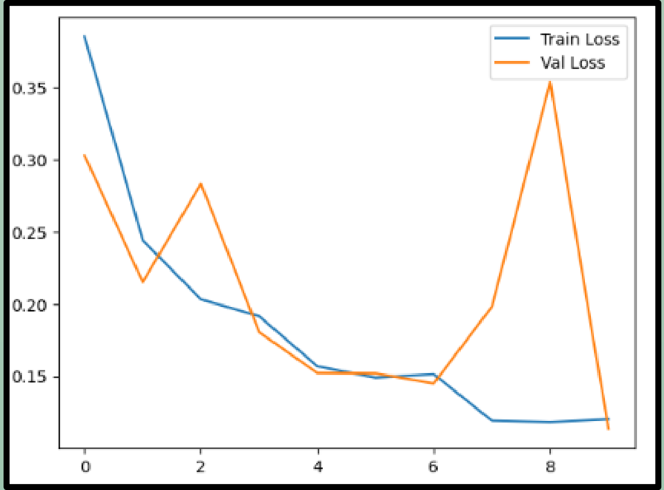

Train Loss
Val Loss
0.9
0.8
0.7
0.6
0.5
0.4
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Confusion Matrix
True Label
Predicted Label
0 1 2 3 4 5 6 7 8 9
0: 523 1 12 3 0 4 24 0 13 4
1: 3 560 14 0 0 16 0 0 4 2
2: 1 2 593 3 1 8 11 2 2 1
3: 0 1 3 468 5 0 2 2 5 2
4: 0 3 1 12 468 0 6 15 7 0
5: 2 1 14 1 0 374 3 0 11 1
6: 19 0 64 5 19 1 376 2 7 0
7: 0 4 3 28 4 0 0 551 0 0
8: 5 3 3 23 0 7 1 0 453 0
9: 15 13 3 1 0 4 0 4 8 560
500
400
300
200
100
0

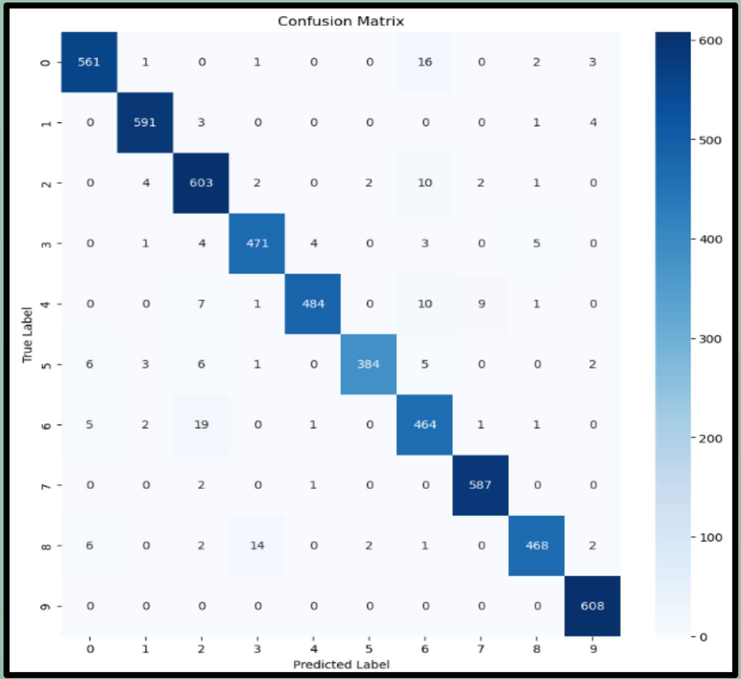
CNN Literature Comparison & Model Results

- ❑ Based on "Introducing EuroSAT"
  - ➢ Tested ResNet50, GoogleNet, BoVW
  - ➢ RGB images performed best
- ❑ CNNs shown to work well for land classification (Cheng et al., 2017)
- ❑ We used pretrained ResNet18
  - ➢ 96.58% accuracy after 10 epochs
  - ➢ Low validation loss, strong generalization
- ❑ Literature supports our method
  - ➢ "Transfer learning from ImageNet significantly boosts performance" – Zhu et al., 2017
- ❑ CNN features transfer well to remote sensing
  - ➢ "Features from object images can be transferred to satellite imagery" – *Penatti et al., 2015*

```
Epoch:  1
Train Loss:  0.3855838748198506
Val Loss:  0.3031127001916946
Val Accuracy:  89.55555555555556
Epoch:  2
Train Loss:  0.24425979202764259
Val Loss:  0.2152513220559719
Val Accuracy:  92.85185185185185
Epoch:  3
Train Loss:  0.20332579653257052
Val Loss:  0.2834192048325095
Val Accuracy:  89.88888888888889
Epoch:  4
Train Loss:  0.1916760168654995
Val Loss:  0.18059264459149088
Val Accuracy:  94.03703703703704
Epoch:  5
Train Loss:  0.15697931043954716
Val Loss:  0.152309974525557
Val Accuracy:  94.92592592592592
Epoch:  6
Train Loss:  0.14905013023825311
Val Loss:  0.15201785969872808
Val Accuracy:  95.03703703703704
Epoch:  7
Train Loss:  0.15147783824972608
Val Loss:  0.14505015961219406
Val Accuracy:  95.22222222222223
Epoch:  8
Train Loss:  0.11920313860881268
Val Loss:  0.1983814820820509
Val Accuracy:  93.74074074074075
Epoch:  9
Train Loss:  0.11815525011027332
Val Loss:  0.35382667998241824
Val Accuracy:  89.66666666666667
Epoch:  10
Train Loss:  0.1202878519826619
Val Loss:  0.11372693913967111
Val Accuracy:  96.44444444444444
```



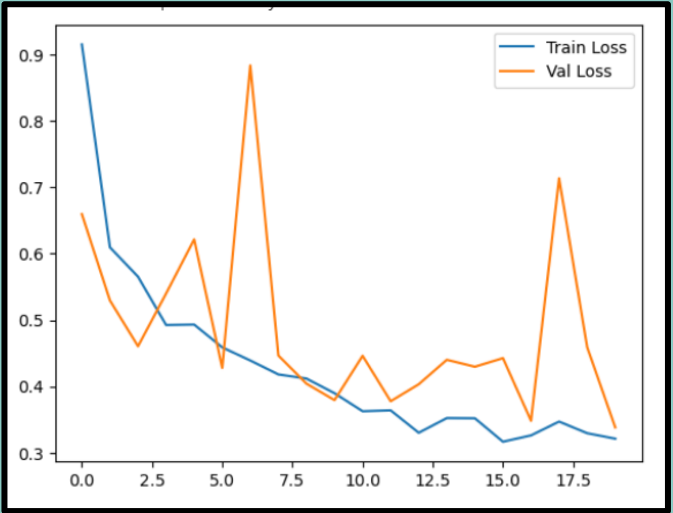

```
Test Loss:  0.10185502828920588
Test Accuracy:  96.68518518518519
```

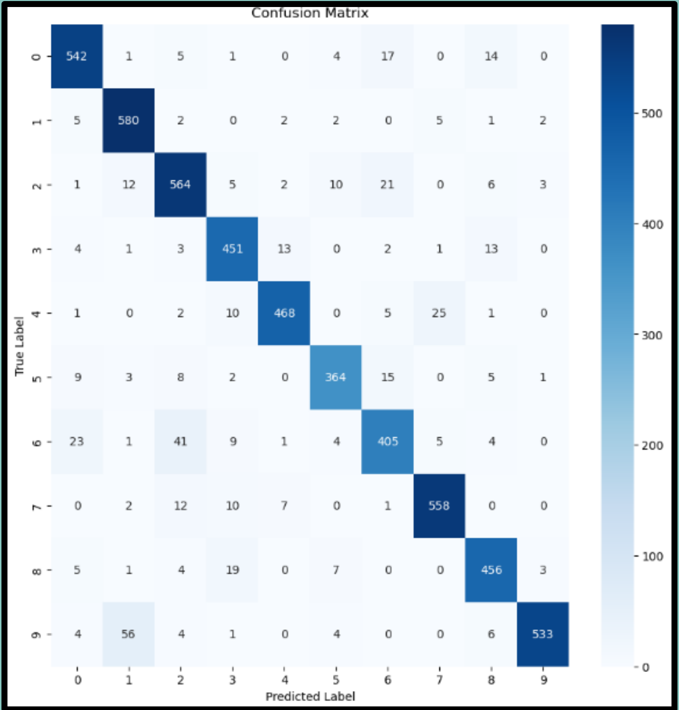


GCN Literature Comparison and Model Results

- ❑ GCNs model data as nodes and edges, useful for graph-structured data
  - ➢ Capture relationships between samples (Chen et al., 2021)
- ❑ We used a GCN model on the EuroSAT dataset
  - ➢ Test accuracy: 91.13%
  - ➢ F1 score: 91.2%
- ❑ Lower accuracy than CNN due to independent images in EuroSAT
- ❑ Research supports GCNs in structured data
  - ➢ “Helped reduce confusion between similar classes” – Li et al., 2020
- ❑ GCN in other studies reached 93.6% accuracy (Chen et al., 2021)
- ❑ GCN worked well, but its strengths depend on dataset structure

```
Test Loss:  0.32498436386094376
Test Accuracy:  91.12962962962963
Test Precision:  0.9126570617405563
Test recall:  0.9112962962962963
Test f1:  0.9112251986135149
```





Conclusion

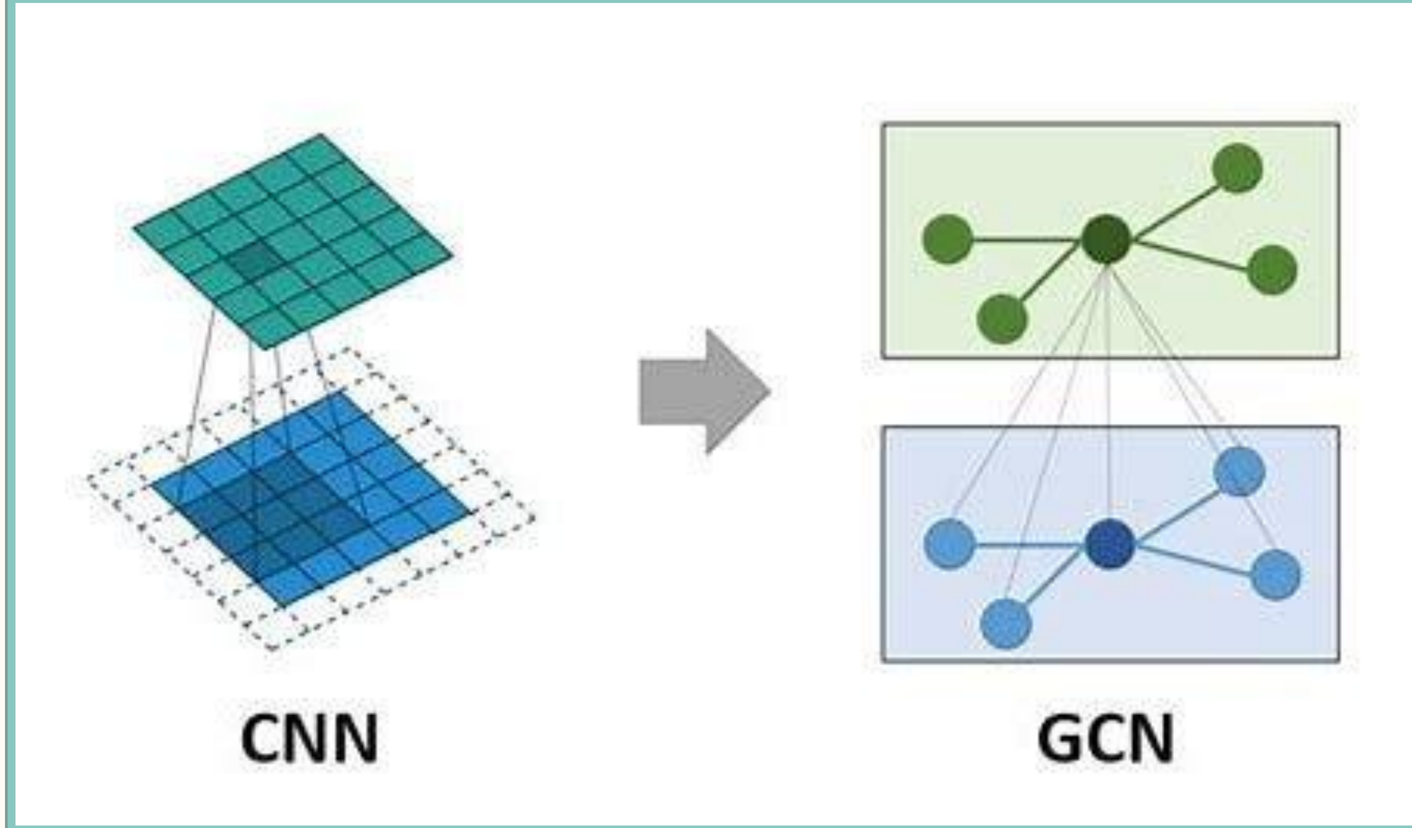
- Compared CNN (ResNet-18) and GCN for EuroSAT satellite image classification

- ResNet-18 achieved higher accuracy (96%) and consistent performance across all classes

- GCN captured relational context but was sensitive to graph structure and connections

- Classification errors in GCN were more common among visually similar land types

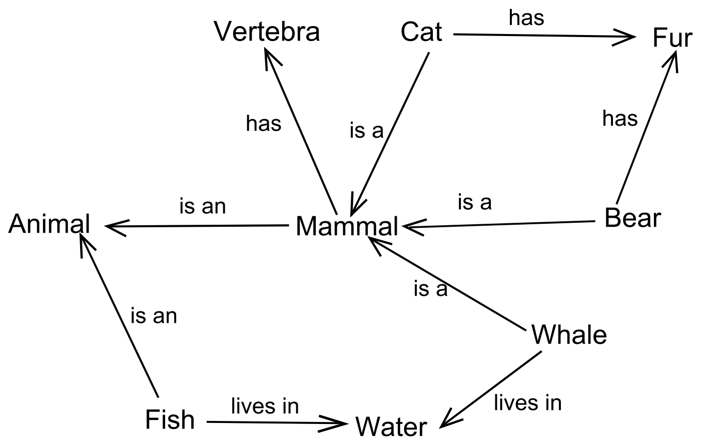
- ❑ Results highlight the strength of CNNs for image-based tasks and the potential of GCNs with improved graph design

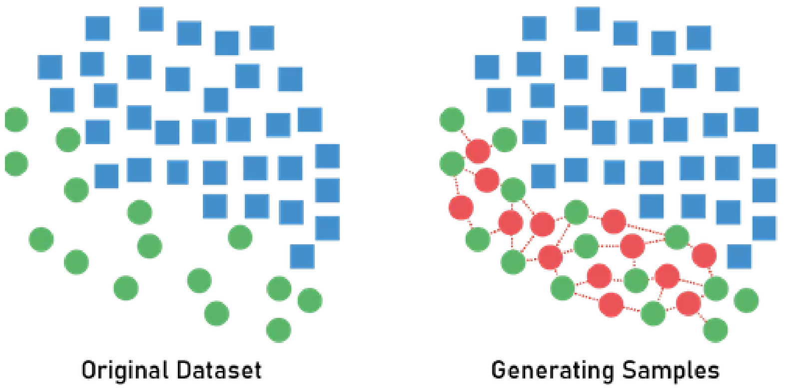


Limitations and Future Works

- Batchwise k-NN graphs may not capture global relationships
- Geographic coordinates of image patches are ignored
- Shallow architecture (2 layers) may not capture complexity

Limitations and Future Works (Cont.)





- Enter Global Spatial-Semantic Graphs. Merges three works
  - ResNet-18 extracted features
  - Semantic Similarity
  - Geographic Proximity
- Graph Attention Networks (GAT)
  - Stabilize learning
  - Combats vanishing gradient
- Address class imbalance
  - Data augmentation techniques
  - Synthetic Minority Oversampling Technique (SMOTE)

Citations

- Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification.
- 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.
- 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
- 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
- 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.
- 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/.
- 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.
- Huang, Liwei , et al. "IEEE Xplore Full-Text PDF": *Ieee.org*, 2025, ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10330561. Accessed 13 Apr. 2025.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). Graph Attention Networks. *ArXiv*. https://arxiv.org/abs/1710.10903