Assessing the Performance of Progressive Fusion in Computer Vision for Land-Use and Landcover (LULC) Applications

By Lisa Liubovich

December 5, 2024

DATA-642: Advanced Machine Learning

Special Thanks To

Professor Peder V. Nelson	NASA	National Oceanic and Atmospheric Administration (NOAA)	National Science Foundation (NSF) and US Department of State
For his guidance and mentorship	Sponsor	Supporter	Supporters

TABLE OF CONTENTS

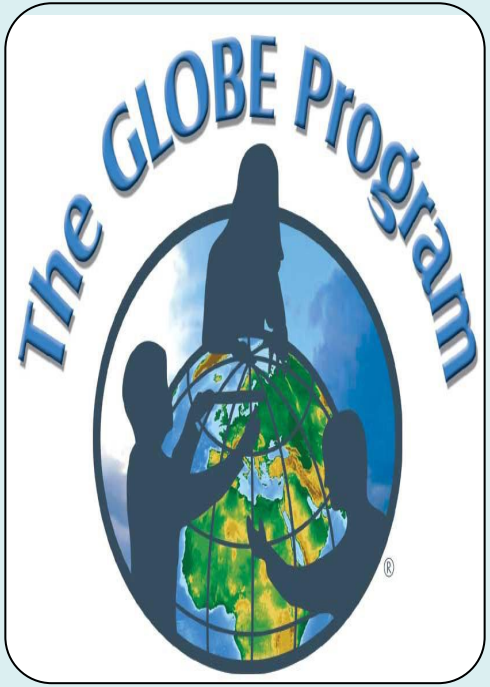
Introduction

What is the GLOBE Program? Why does it matter? What is progressive fusion?

GLOBE Program

What is it and why is it important?

- **What is GLOBE?**
  - **Global Learning and Observations to Benefit the Environment (GLOBE)** → international science and education program established in **1995**
  - Launched the **GLOBE Observer mobile app** to increase spatial and temporal coverage of GLOBE data (**2020**) → "citizen scientists" report ground based observations
- **Why is it important?**
  - **Ground-based observations** of environmental data are critical to the **interpretation/downscaling of satellite products**
  - **In-situ validation of land-use and land-cover (LULC) products** play an important role in **improving the accuracy of models employing remotely sensed data** and **map products for practical management purposes**



Progressive Fusion

What is progressive fusion?

- **Inspiration:** Huang, X., Yang, D., He, Y., Nelson, P., Low, R., McBride, S., Mitchell, J., & Guarraia, M. (2023). Land cover mapping via crowdsourced multi-directional views: The more directional views, the better.
- **What is progressive fusion?**
  - Iteratively applying fusion at each layer with backprojective connections, where fused multimodal features **continuously inform the encoder throughout processing**, resulting in richer feature refinement at multiple levels.

Hypothesis: Progressive fusion will help improve classification accuracy by improving how multimodal data is fused in deep learning models, specifically those with encoder-decoder architectures

Data and Methodology

Data

- Data obtained via GLOBE API at the end of September
- Cleaned in R for completeness, added seasonality variables
- Other experiments done:
  - Classifiers → k-means and DBSCAN Random Forests
  - LASSO Regression
  - CCA → found that digital attributes and surface conditions are not strongly correlated
    - **This will become important later**

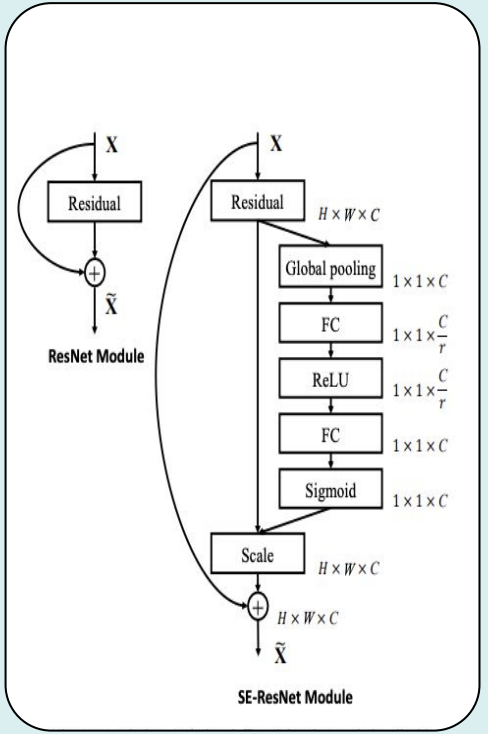
Classifier	Snow_ice accuracy	Leaves_on_trees accuracy
K-means random forest	0.9551	0.9909
DBSCAN random forest	0.9983	0.9963
LASSO (digital attributes	0.9655	0.8621

Table of Previous Experimental Results

Methodology

Section 1: Encoder-Decoder Architecture

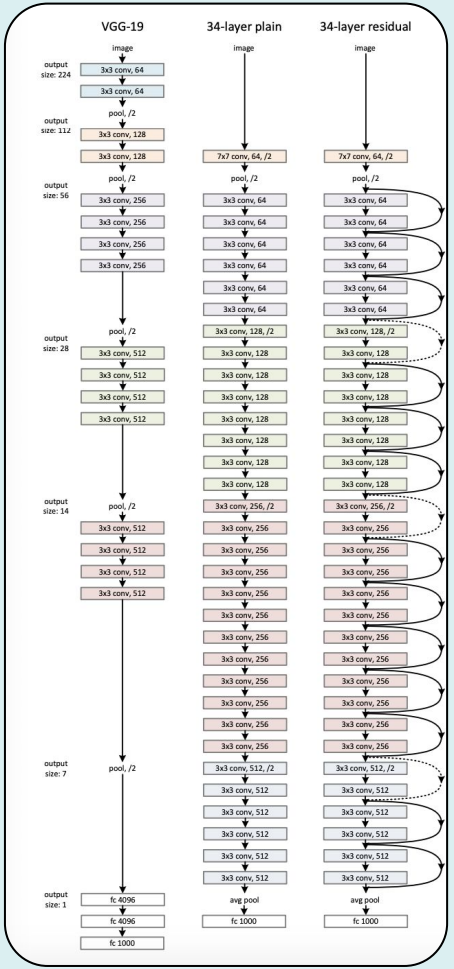
- **ResNet/CNNs:** Using ResNet's standard convolutional layers and residual blocks, which are organized in stages. Each stage has progressively increasing channels, allowing deeper feature hierarchies.
- **SE blocks:**
  - **Squeeze (Global Average Pooling)**
  - **Excitation (Fully Connected Layers)**
  - **Reweighting (Scaling)**
- **Skip Connections:** Utilize ResNet's skip connections to maintain gradient flow across layers, critical for deep networks. These skip connections also facilitate transferring spatial information from the encoder to the decoder.



Methodology

Section 1: Encoder-Decoder Architecture

- **Upsampling:** Used transposed convolutions or nearest-neighbor upsampling followed by convolution layers for each decoder stage, increasing the spatial resolution at each step.
- **Feedback Loops:** To support progressive fusion, add backprojective connections that iteratively improve feature refinement:
  - **Fusion Layers:** After each decoder stage, fused multimodal features (e.g., text and image) are backprojected into the encoder layers through feedback connections.
  - **Impact on Encoder:** Each backprojected fused representation guides the encoder's feature maps with late-stage multimodal context, iteratively refining early representations.
  - **Progressive Refinement:** This structure supports a feedback loop in which the encoder benefits from the fused multimodal information, yielding more robust representations for final segmentation.



Methodology

Section 2: Fusion Strategy

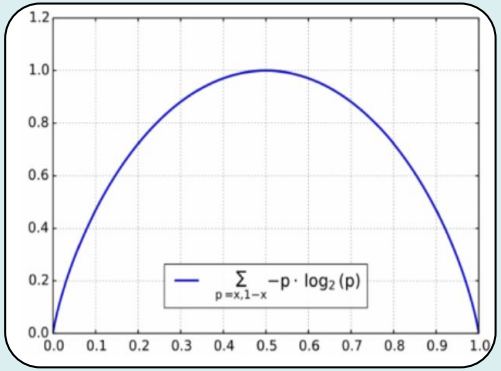
- **Early Fusion (Baseline):** Concatenate features from different modalities (e.g., text and image) at the beginning of the encoder, effectively combining modalities before deeper feature extraction.
- **Late Fusion (Baseline):** Fuse multimodal features just before entering the decoder, allowing the network to process each modality separately through the encoder.
- **Progressive Fusion (Proposed):** Iteratively apply fusion at each layer with backprojective connections, where fused multimodal features continuously inform the encoder throughout processing, resulting in richer feature refinement at multiple levels.

Example of image collage used as input data

Methodology

Section 3: Loss Functions

- **Binary Cross-Entropy (BCE) Loss:** This loss function measures the pixel-wise accuracy of segmentation, treating each pixel as a binary classification task.
- **Dice Loss:** Dice Loss focuses on the overlap between the predicted segmentation map and ground truth, targeting global shape accuracy. It is particularly effective for segmenting objects with variable sizes.
- **Combined Loss:** Use a weighted sum of BCE and Dice Loss, optimizing the model for both fine pixel-wise accuracy and accurate overall shape, enhancing both local detail and general segmentation quality.



Methodology

Section 4: Training Strategies, Hyperparameters, and Environment

- Split: 80/20
- Python version: python 3.10.15
- Libraries used: TensorFlow, NumPy, SKLearn, Concurrent, Pandas, MatPlotLib,
- Computer/GPU: Apple M3 Max, 30 cores, Metal 3 support
- Optimizers: adam, RMSprop
- Learning rate(s): starting with 0.00001, then with learning rate schedulers
- Epochs: 20
- Early stopping and recording method
- Performance evaluation metrics:
  - Validation accuracy
  - Validation loss
  - AUC

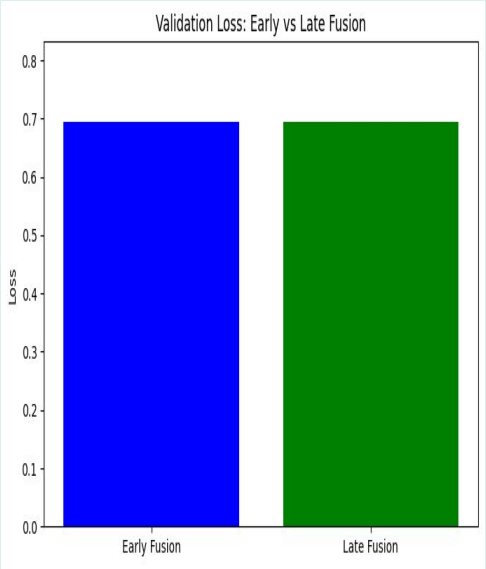
Results

Early vs. Late. vs. Progressive Fusion (tuned)

Baseline Results

Early and Late Fusion

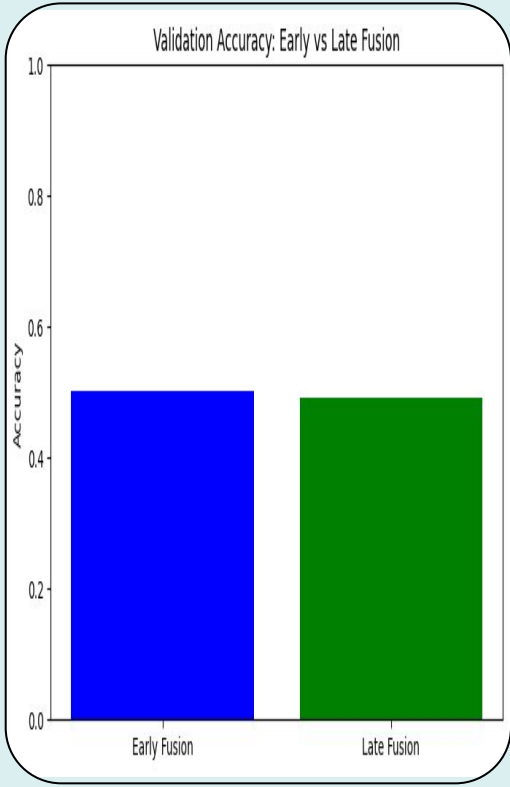
- **Loss (0.693) is High:** A loss of ~0.693 is typical for binary classification with poor predictions (close to random guessing). This indicates that **neither model is effectively learning the relationship between input features (images and metadata) and labels.**
- **Accuracy (~50%) Matches Random Guessing:** Both models are performing no better than random guessing (50% for binary classification). There may be issues with the data, the models, or the fusion approach.



Baseline Results (cont 'd)

Early and Late Fusion

- **Cross-Modal Interaction Limits:**
  - **Early Fusion** May suffer from overfitting if the two modalities are not strongly correlated/might struggle to learn distinct representations for each modality.
  - **Late Fusion** If modalities are weakly informative, the late fusion model might fail to effectively combine the features. Independent processing of inputs could dilute cross-modal relationships.
  - **Both approaches show limited ability to leverage cross-modal information effectively**



Baseline Results (cont 'd)

Label class distributions:

- This means **class imbalance is not the issue**, and the poor performance of the models is likely due to other factors, such as:
  - **Weak Predictive Power:** The image and metadata features may not have a strong relationship with the labels. The current model architecture may not be effective at extracting meaningful patterns.
  - **Overfitting or Underfitting:** The models might be too simple or too complex for the given dataset.
  - **Data Noise or Correlation Issues**: Metadata features might be noisy or weakly correlated with the labels. Images may not provide sufficient discriminative information.

```
Train Class Distribution:
Label 0: 14187 instances
Label 1: 14196 instances

Validation Class Distribution:
Label 0: 3563 instances
Label 1: 3533 instances
```

Preliminary Results

Progressive Fusion

Model Variant	Validation Accuracy	Validation Loss
Early Fusion Model	0.502537	0.693166
Late Fusion Model	0.492531	0.693226
Progressive Fusion Model	0.497182	0.693192
Progressive Fusion (Small Batch) Model	0.5703	0.6872

Methodology

Tuning Strategies

Progressive Fusion

- Reduced learning rate → learning rate schedulers:
  - Reduce learning rate by 0.1 every 5 epochs
  - Cyclical learning rate, oscillating every 5 epochs
- Added dropout regularization layers, reduced SE block reduction ratio, and used L2 regularizer on metadata to address overfitting
  - Attempted multiple dropout values from 0.5 to 0.7
- Changed optimizer from Adam to RMSProp → due to its efficacy with CNNs
- Reduced number of filters → lead to even more severe overfitting
- Hail Mary: try a shallower backbone with L2 regularization, constrained weights, dropout = 0.5
  - Fixed overfitting issue but did not improve model performance significantly

All lead to severe overfitting

Methodology

Final Model Comparisons

Progressive Fusion

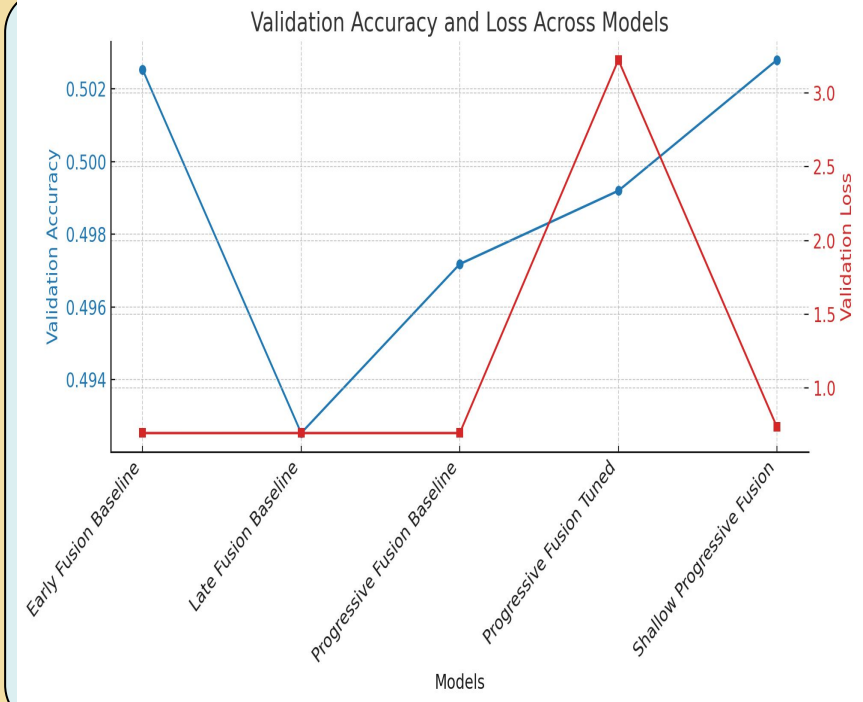
Model	Validation Accuracy	Validation Loss	Training Accuracy	Training Loss	Validation AUC
Early Fusion Baseline	0.502537	0.693166	0.4999	0.6932	NA
Late Fusion Baseline	0.492531	0.693226	0.4956	0.6932	NA
Progressive Fusion Baseline	0.497182	0.693192	NA	NA	NA
Progressive Fusion Tuned	0.4992	3.2244	0.5973	11.5014	0.5049
Shallow Progressive Fusion	0.5028	0.7340	0.5113	0.7252	0.5112

Discussion

Why did we get these results?

Discussion

- **Main Goal:** To evaluate whether progressive fusion with SE blocks outperforms early and late fusion methods.
- **Outcome:** Progressive fusion did not show significant improvement over early and late fusion strategies in accuracy or AUC.
- **Why?**



Discussion

- Possible Reason: **Metadata features may not have been sufficiently informative to boost performance** → plays a crucial role in multimodal fusion tasks
- **if the metadata/features derived from it lacks strong correlation with the target variable, the fusion strategies**—regardless of their complexity—**will struggle to improve performance.**
- Metadata should add complementary information that images alone cannot provide → If the metadata is noisy or weakly correlated, it becomes more of a **distraction** than a helpful input.

Why did we get these results?

Discussion

- **CCA results** → metadata is not strongly correlated to surface condition variables → **the fusion mechanism cannot effectively leverage it.**
- **Also:** the progressive model is so complex that it **requires very finely tuned weights, rendering preloaded weights useless** → did not have time to use exclusively weights from scratch

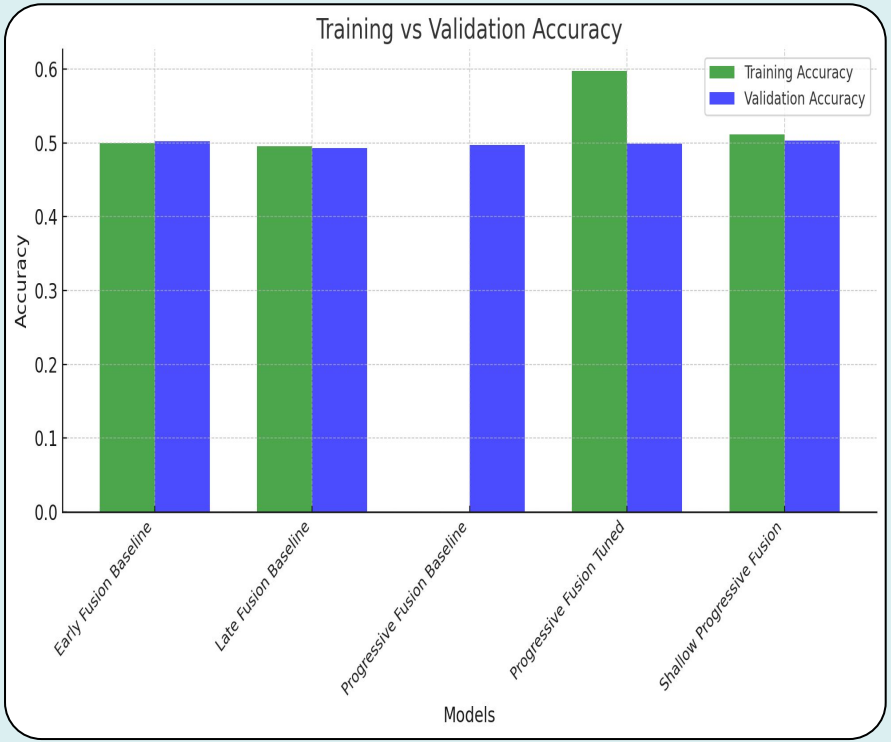
Now what …

Future Recommendations and Conclusions

Next steps, possible solutions

Future Recommendations and Conclusions

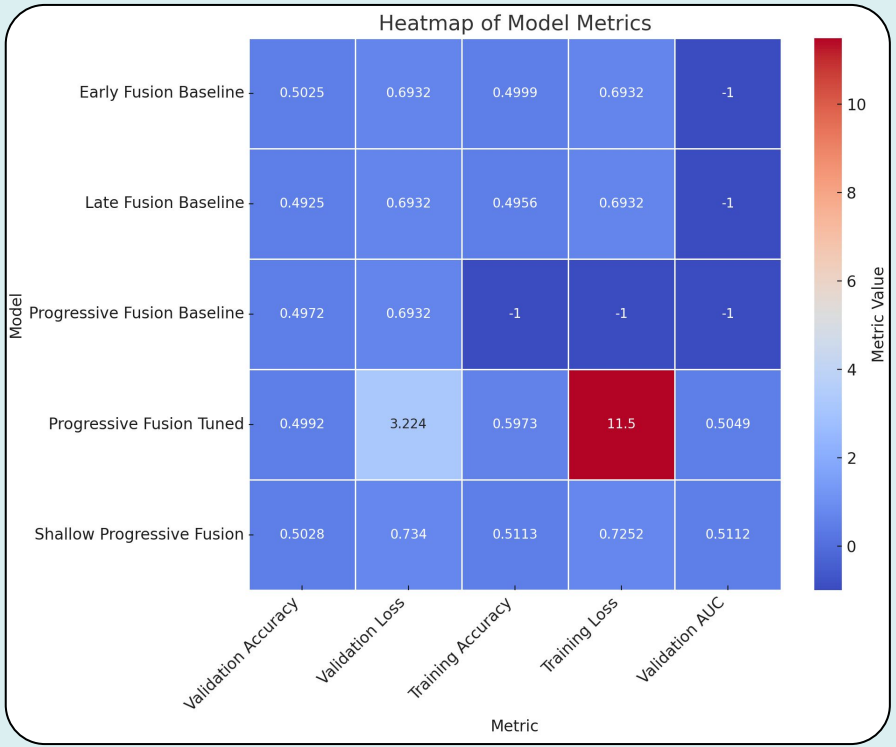
What now?



1. **Improve Metadata Quality with New Features** → Explore additional digital attributes that might correlate more strongly with surface conditions
2. **Feature Engineering** → Apply transformations to derive richer metadata features

Future Recommendations and Conclusions

What now?



3. **Simpler Fusion Tasks** → Test fusion strategies on simpler, synthetic tasks where the correlation between metadata and labels is stronger. This would allow us to isolate and evaluate the effectiveness of the fusion mechanisms themselves.

4. **Weighting Metadata Contributions** → Use attention mechanisms to dynamically weight the contribution of metadata during fusion. This could help the model "ignore" irrelevant metadata when it's not helpful.

SUMMARY

1. Progressive fusion did not improve classification accuracy of GLOBE ground photos

2. Why? Because the metadata/digital attributes are not highly correlated with the surface condition variables.

3. Next Steps: improve metadata with new features, feature engineering, simpler fusion tasks, and weighting metadata contributions

THANK YOU