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

By Lisa Liubovich

Introduction

The increasing availability of satellite imagery and ground-based observations has revolutionized land-cover and land-use classification, enabling more detailed and accurate environmental monitoring. The Adopt a Pixel initiative, part of the GLOBE Program, empowers citizen scientists to contribute valuable ground-truth data through the GLOBE Observer app. Despite these advancements, a significant challenge remains: the need for accurate in-situ validation to enhance the performance of remote sensing models. This gap has spurred interest in developing deep learning-based fusion strategies to combine multimodal data (e.g., imagery and metadata) effectively. Current research explores various fusion approaches, including early, late, and progressive fusion, to integrate multimodal features and improve classification accuracy. Early fusion merges data at the input level, while late fusion combines features at the end of the network. Progressive fusion, a more recent approach, iteratively integrates multimodal features throughout the network using backprojective connections. This study aims to evaluate these fusion strategies within an encoder-decoder framework featuring Squeeze-and-Excitation (SE) blocks, using data from the GLOBE API. The primary objective is to determine which fusion strategy most effectively predicts surface conditions, such as snow and leaves on trees, and to understand the limitations and potential improvements for multimodal fusion in land-cover classification tasks.

Literature Review

The Adopt a Pixel Project is part of the GLOBE Program, or Global Learning and Observations to Benefit the Environment (GLOBE), an international science and education program established in 1995. The GLOBE Observer app launched in 2020, with aims to increase spatial and temporal coverage of GLOBE data. The app allows “citizen scientists” to report ground based observations of clouds, land cover, mosquito habitats, and/or tree height (Low et. al, 2021). Low et. al. (2021) emphasizes how important ground-based observations of environmental data for the interpretation and downscaling of satellite products as well as the role of in-situ validation of land-use and land-cover (LULC) in improving the accuracy of remote sensing models and map products for practical management applications. The notion of addressing this gap in in-situ validation of LULC with deep learning models has gained significant traction since the early 2020s. Huang et. al. (2023) echoed the concerns of Low et. al. (2021), contending that the lack of sufficient ground truth and training samples is a major challenge in accurately providing land cover information, especially in economically-distressed countries. One proposed solution to this challenge is an end-to-end classification task using deep convolutional neural networks to map complex non-linear relationships, notably using four different views and three fusion strategies (early, late, and score fusion in particular) in an attempt to maximize classification accuracy (Huang et. al, 2023). Huang et. al. (2023) found

success in EfficientNet as deep learning architecture and late fusion, concluding that the more directional views included in the data the better, which is best implemented in later stages of deep learning architectures since fusing multi-directional views at early stages might confuse the models.

Yuan et. al. (2023) alternatively proposed a new encoder-decoder semantic segmentation network MPFFNet, based on the DeepLabv3+ semantic segmentation model and drawing on multipath feature fusion. MPFFNet was found to perform better than classical semantic segmentation networks like U-Net, PSPNet, and DeepLabv3+ in both large-scale classification and fine land-cover classification settings (Yuan et. al. 2023). Another proposed solution is the Multi-directional and Multi-constraint Learning Network (MMLN) to address the challenges of inter-class homogeneity and small objects in remote sensing imagery, centered around a decoder structure called the Multi-directional Dynamic Complement Decoder (MDCD) (Sun et. al., 2024). In addition, Sun et. al. (2024) also proposed the Multi-Constraint Saliency Boundary-adaptive Module (MSBM), which refines the model's precision in segmenting small objects. The combination of these networks and modules resulted in a model that outperformed state of the art methods on multiple datasets (Sun et. al. 2024). Atef et. al. (2023) leveraged an integrated approach, combining spatial models such Artificial Neural Network (ANN), cellular automation (CA) model, Conversion of Land Use and its Effects (CLUE) model, and Hybrid Cellular Automata Markov Chain (CA-MC). This was further integrated with the Multi-Layer Perceptron Neural Network (MLP-NN), as MLP-NN facilitates the automatic calibration of the CA-MC model and provides a more accurate future change scenario than CA-MC alone. (Atef e. al. 2023). The Swin-radial-locality network (SRLN) was also found to outperform other state of the art, showing promising and uniform performance across a spectrum aerial altitudes (Lv et. al. 2024). Similarly, Xiao et. al. (2024) proposed an MFEPNet architecture using ResNet50 and Swin Transformer backbone networks to capture multi-scale features, which also outperformed other models such as RefineNet and CTCNet. These methods are considered Pixel-Based Mapping Methods, which is a common strategy for landcover mapping tasks that do not require strict delineation of target boundaries; a common drawback, however, is that these methods can suffer from reduced resolution of the landcover map (Qin & Liu, 2022). This can be problematic, as the resolution reduction will likely cause a loss of remote-sensing image resolution and spatial information (Yuan et. al. 2023). In order to combat this problem, Qin & Liu (2022) identify Object-Based Image Analysis as an alternative.

Scholars also frequently advocate for research combining multiple advanced techniques, such as leveraging both CNNs and Transformers (Sun et. al. 2024, Yuan et. al. 2023) or incorporating transfer learning (Qin & Liu, 2022, Huang et. al. 2024). Alharbi et. al. (2023) proposed a data augmentation method to combat overfitting and improve performance called Quality Based Sample Selection (QSS), which outperformed state of the art methods by increasing the quantity, diversity, and quality of training data. Finally, Qin & Liu (2022) identified several different fusion approaches, such as pixel-level fusion such as in PCA, feature-level fusion such as in Recurrent Neural Networks (RNNs) and CNNs, decision-level

fusion, and multi-view fusion. Similarly, Huang et. al. (2023) proposes progressive fusion to mitigate the information loss in late fusion and the feature heterogeneity issue in early fusion, with a major caveat: Progressive fusion requires high-level modification of deep learning architectures, leading to the uselessness of pretrained weights. It is this emerging fusion strategy that inspired the motivation for this paper.

Method

Data, Preprocessing, and Preliminary Analyses

The data used in this paper was obtained from the GLOBE API, containing data ranging from 1995 to late September 2024 (Global Learning and Observations to Benefit the Environment (GLOBE) Program, 2024). Data was thoroughly pre-processed in Rstudio using R 4.4.1, standard tidyverse libraries, and the LandsatLS R package to extract satellite data from Google Earth Explorer. After cleaning for completeness, the final cleaned dataset contained 35,507 observations, each containing numerical variables such as date, latitude and longitude coordinates, surface conditions, and six different photo urls representing six directions (north, south, east, west, upward, and downward). Data versions and preprocessing code were uploaded to both Huggingface and Github repositories for reproducibility. The majority of the data preprocessing process was focused on building classifiers to predict the presence of certain surface conditions, namely snow and ice as well as the presence of leaves on trees. Table 1 below summarizes the classifiers attempted on numerical data, using the entire dataset:

Table 1

Classifier	**Hyperparameters**	**Validation Accuracy for snow_ice**	**Validation Accuracy for leaves_on_trees**
Logistic regression (using k-means cluster labels as features)	K = 8 (using elbow method)	0.9502	0.9016
K-means clustering Random Forest (RF)	K = 8 Split: 80/20 Number of trees = 100	0.9531	0.9873
K-Means RF (Tuned)	K = 8 Split: 80/20 Mtry = 3 (using k-fold cross validation with k = 5) Number of trees = 500	0.9551	0.9909

DBSCAN Clustering RF	Eps = 0.5 (using KNN plot) minPTs = 10 (roughly twice the number of surface condition features) Split: 80/20	0.9983	0.9963

Both the K-means RF (Tuned) and DBSCAN Clustering RF had barely higher or lower training validation accuracies for each surface condition, indicating that neither classifier suffered from overfitting. Following the success of these last two classifiers, the focus then shifted to applying a similar preprocessing process to the digital attributes found in each of the six directional views of a location. After examining the shape and spread of digital attributes such as brightness, chromatic bands, and color magnitude, several different methods of analysis were attempted to build a similarly powerful classifier. Both clustering RF methods, regardless of tuning strategies, had accuracy rates of 100%, indicating severe overfitting and over-complexity in model selection. A simpler logistic regression model was attempted but did not converge due to perfect multicollinearity. Finally, a LASSO regression found solid accuracy rates but evidence of class imbalance; in order to address this, both upsampling and downsampling were attempted, but both approaches yielded significantly lower accuracy. The final LASSO regression model using n = 100, unlike the numerical attribute classifiers (due to computational expense of extracting metadata) saw an accuracy rate of 0.9655 for snow_ice and 0.8621 for leaves_on_trees. The experiment was reattempted using an n = 200, with an accuracy of 0.9650 for snow_ice and 0.8600 for leaves_on_trees, with no improvement or decline after tuning lambda values and prediction thresholds.

Finally, a canonical correlation analysis was performed to try to make sense of the disconnect between digital attributes and surface condition variables. The CCA found weak to moderate association between the first linear combination of variables in each set (0.2195816) and weak association (0.1660211) for the second pair of linear combinations. Moreover, the CCA found a small negative impact of brightness on the first canonical variate, small positive impact of the red color band mean value, and a relatively more significant impact of green_mean on the first variate for X, while snow_ice is the more influential variable for Y. This lack of correlation between digital attributes and surface condition variables becomes critical in explaining the results of the fusion strategy experiment.

In order to prepare the data for training in deep learning contexts, the six pictures contained in each of the 35,507 observations were concatenated together in a “film strip mosaic” (example seen in Appendix). Almost all rows of images were successfully converted into mosaic images, a process that was done in Rstudio over the course of roughly three hours. In order to maximize computational efficiency, images were resized to 250 x 188 and processed in batches of 1000 using parallel processing. These images ultimately served as the input into the encoder.

Fusion Strategy Experiment

Encoder-Decoder Structure

The model assessed in this paper follows a widely used encoder-decoder structure, with Squeeze-and-Excitation (SE) Blocks as introduced by Hu et al. (2017). The ResNet architecture was introduced in 2015 in order to address a common issue with deep neural networks: degradation (He et. al. 2015). Deep networks themselves naturally integrate features of various levels and classifiers next to and on top of each other; the "levels" of features can be enriched by the number of stacked layers, or depth. By 2015, depth was established as critically important for both image processing tasks and non-trivial visual recognition tasks (Dicarlo et. al. 2012, Chen & Cowan 2013, Tsotsos et. al. 2008). However, simply stacking more layers is notorious for "vanishing" gradients, as adding additional layers to a deep model leads to a higher training error. This degradation is addressed by a deep residual learning framework, where the layers explicitly fit a residual mapping. Specifically, if H(x) is an underlying mapping to be fit with a few stacked layers, we can let these layers approximate a residual function $\mathcal{F}(\mathbf{x}) = \mathcal{H}(\mathbf{x}) - \mathbf{x}$; The original mapping is recast into $\mathcal{F}(\mathbf{x}) + \mathbf{x}$ The Encoder in this paper uses ResNet's standard convolutional layers with mostly 3 x 3 filters and residual blocks $y = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}$ applied to every few stacked layers, where X and y are the input and output vectors of the layers considered and $\mathcal{F}(\mathbf{x}, \{W_i\})$ represents the residual mapping to be learned. These convolutional layers and residual blocks are organized in stages with progressively increasing channels, allowing for deeper feature hierarchies (He et. al. 2015).

The approach explored in this paper integrates Squeeze-and-Excitation (SE) Blocks, an architectural unit proposed by Hu et. al. (2017) to improve the quality of representations produced by a network by explicitly modeling the interdependencies between the channels of its convolutional features. For any given transformation $\mathbf{F}_{tr}$, mapping the input $\mathbf{X}$ to the feature maps $\mathbf{U}$ where $\mathbf{U} \in \mathbb{R}^{H \times W \times C}$, we can construct a corresponding SE block to perform feature recalibration. Features U are first passed through a squeeze operation, which produces a channel descriptor by aggregating feature maps across their spatial dimensions (H x W) via global average pooling, which is a structural regulator that prevents overfitting for the fully-connected layers and is more robust to spatial translations of the input (Hu et. al., 2017, Lin et. al., 2013). The excitation block uses a parameterized gating mechanism $\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \sigma(g(\mathbf{z}, \mathbf{W})) = \sigma(\mathbf{W}_2\delta(\mathbf{W}_1\mathbf{z}))$ with sigmoid and ReLU activation to limit model complexity and help with generalization. The SE blocks allow the network to focus on features that enhance segmentation accuracy by multiplying each channel in the feature map by its learned weight to emphasize more important channels and downscale less important ones (Hu et. al. 2017). The Encoder also utilizes ResNet's skip connections to maintain gradient flow across layers and facilitate the transfer of spatial information from Encoder to Decoder (Goodfellow et. al 2016).

The decoder essentially follows a structure that mirrors the encoder in reverse. It starts with the highest-level (most abstract) feature maps generated by the encoder and works backward, progressively increasing the spatial resolution at each stage. Mirroring the encoder

structure allows the decoder to gradually reintroduce spatial detail. This process helps the network move from a condensed representation back to a fully detailed image, which is crucial for tasks like image segmentation, where pixel-level accuracy is important (Sellat et. al., 2022, Goodfellow et. al., 2016). SE blocks are then inserted after each upsampling operation in the decoder. After each upsampling step, SE blocks dynamically reweight the channels, adjusting the attention to fit the new spatial resolution. As the feature maps get larger, different parts of the image (or different features) might need more or less emphasis depending on the spatial context. (Hu et. al., 2017). Each set of layers in ResNet learns what's called a "residual", which is the difference between the original input and the features learned by that set of layers. Mathematically, instead of just learning $y = f(x)$, it learns $y = f(x) + x$, where x is the input to the set of layers (which comes from an earlier part of the network), $f(x)$ is the transformation applied by that set of layers, and y is the output that now includes both learned features $f(x)$ and original input x. This way, if the network cannot learn new features that improve the output, it can "skip" that layer by just passing x through, keeping the information flowing smoothly (Sellat et. al., 2022, Goodfellow et. al., 2016).

To support progressive fusion, backprojective connections were added to iteratively improve feature refinement. This is accomplished through feedback loops, which are mechanisms that take outputs from one part of the network and feed them back into earlier parts. In this architecture, feedback loops are established through backprojective connections, which direct information from the decoder back into the encoder. After each decoder stage, the processed feature maps are sent back to specific encoder layers. This feedback loop is continuous, meaning that the encoder continually receives updated information as the decoder progresses through its stages. This iterative flow of information is what enables progressive refinement of the encoder's features. By using feedback, the network can continuously improve its understanding of the input, allowing early layers to gain insights that normally only deeper layers would provide. This structure results in more coherent and context-aware feature maps in the encoder, which leads to more accurate segmentation or classification results (Talmi et al., 2017, Bertinetto et al., 2016). Fusion layers integrate multimodal features (like text and image) by merging information from different modalities at each stage of the decoder. After each decoder stage, the feature maps, which may include fused representations from multiple sources, are "backprojected" (or fed back) to corresponding layers in the encoder. Fusion layers act as intermediaries that prepare this information for feedback by ensuring it aligns with the encoder's requirements. This process maintains a consistent flow of multimodal context throughout the network, enriching the encoder's feature maps with information beyond what was initially provided. For tasks involving multiple types of data, like image-text fusion, the fusion layers enhance the encoder's ability to capture complex relationships. Each fusion layer adjusts the content being backprojected so that it directly benefits the encoder's features, resulting in a more nuanced representation (Rozental & Biton, 2019, Dong et al., 2016).

The backprojected information acts as additional context for the encoder, allowing it to refine its feature maps based on the information generated at later stages in the decoder. Each

backprojected representation provides the encoder with a "preview" of what the network is learning at deeper levels. With each new backprojected feature map, the encoder can re-evaluate and enhance its earlier feature representations. This continuous guidance effectively makes the encoder layers more adaptive, improving the quality of the features they produce by incorporating insights learned during later processing stages. This iterative refinement ensures that early-stage representations are not static; instead, they evolve to incorporate more context, leading to more expressive and accurate final output, which is particularly valuable for complex segmentation tasks (Lin et al., 2017, Chen et al., 2018).

Fusion Strategies

Early fusion is a strategy where features from different modalities are combined at the beginning of the encoder. In this setup, all multimodal data (e.g., text and image features) are concatenated or merged before the network's deep feature extraction layers.This strategy takes the input data from each modality and joins it into a single, unified representation. The concatenated features are then fed into the encoder, allowing the network to process them together from the very first layers. By merging all data upfront, early fusion enables the network to learn joint feature representations from the start, which can help it find simple cross-modal patterns quickly. This approach limits the flexibility to adaptively combine information at different stages. The network processes the combined features as a single input, which may prevent it from capturing complex interactions that emerge only at deeper levels of representation (Brabandere, 2024, Pulapakura, 2024).

Late fusion combines features from different modalities at a later stage, usually just before the decoder. In this setup, the encoder processes each modality separately, keeping the features distinct until they are combined in the fusion step. Each modality has its own processing path within the encoder, which allows the network to extract specific, modality-dependent features. Just before the data enters the decoder, the features are fused, usually through concatenation or addition. Each modality has its own processing path within the encoder, which allows the network to extract specific, modality-dependent features. Just before the data enters the decoder, the features are fused, usually through concatenation or addition. By keeping the modalities separate until late in the network, this strategy may miss out on potential interactions and correlations between the modalities at earlier stages, which could be useful for tasks where the modalities are interdependent (Brabandere, 2024, Pulapakura, 2024).

Progressive fusion is a strategy that combines features from different modalities iteratively at each layer of the encoder and decoder. This fusion approach involves continuously integrating multimodal information with backprojective connections that feed information from the decoder back to the encoder. With progressive fusion, the network does not just fuse the modalities at the beginning or end. Instead, it integrates multimodal features at multiple points throughout the network. Each layer has the opportunity to incorporate insights from other modalities, which allows features to evolve in a multimodal context. Backprojective connections

ensure that feedback from deeper layers informs earlier layers, continuously refining the multimodal representations (Sellat et. al., 2022). Progressive fusion allows for a dynamic and continuous exchange of information between modalities, resulting in more contextually enriched features. Since multimodal information is incorporated at each layer, the network can capture intricate relationships and dependencies across modalities. This approach is especially valuable for tasks that require a high level of detail and nuanced feature interactions, such as segmentation or complex object recognition. Progressive fusion is more computationally intensive, as it requires additional connections and processing to continuously integrate information across layers. However, the improved feature refinement typically justifies the additional complexity in tasks that benefit from nuanced multimodal understanding (Brabandere, 2024, Pulapakura, 2024).

Loss Functions

The encoder-decoder structure, combined with SE blocks, enables the model to capture detailed spatial information at each layer. Binary Cross Entropy (BCE) Loss encourages precise classification at the pixel level, leveraging the rich spatial detail provided by the decoder with SE-enhanced features. It is a widely used loss function for image segmentation tasks and works best in equal data distribution among classes scenarios. The BCS Loss function is defined as $L_{BCE}(y, \hat{y}) = -(y\log(\hat{y}) + (1 - y)\log(1 - \hat{y}))$, where $\hat{y}$ is the predicted value by the model (Jadon, 2020). The use of Dice Loss (DL) reinforces the structure of segmented objects, ensuring that the encoder's progressively refined features maintain the correct shape as information flows through backprojective connections. This is critical as the encoder receives feedback from deeper layers, progressively refining its understanding of object shapes. The DL is often used in image classification tasks and is defined as $DL(y, \hat{p}) = 1 - (2y\hat{p} + 1)/y + \hat{p} + 1$, where 1 is added in the numerator and denominator to ensure that the function is not undefined in edge case scenarios such as when $y = \hat{p} = 0$ (Jadon, 2020, Sellat et. al., 2022). The final loss function used in this model is Combined Loss (CL), which helps the network balance detailed feature extraction from individual modalities with the coherent integration of multimodal information. This balance is essential to achieving accurate segmentation across various objects, even with complex shapes and mixed input data. The CL uses a weighted sum of BCE and DL, optimizing the model for both fine pixel-wise accuracy and accurate overall shape, enhancing both local detail and general segmentation quality. The CL is defined as $L_{m\text{-}bce} = -1/N \Sigma_i \beta (y - \log(\hat{y}) + (1 - \beta)(1 - y)\log(1 - \hat{y})$ and $CL(y, \hat{y}) = \alpha L_{m\text{-}bce} - (1 - \alpha)DL(y, \hat{y})$, where DL is Dice Loss (Jadon, 2020).

Training Strategies, Hyperparameters, and Environment

All deep learning approaches were done in python 3.10.15 using standard data visualization and machine learning libraries such as TensorFlow, NumPy, SKLearn, Concurrent, Pandas, and MatPlotLib. All training was conducted on the Apple M3 Max with 30 cores and Metal 3 support. Roughly 80% of data was used as training data while approximately 20% was reserved for validation. The Adam optimizer was first used for every model before switching to

RMSprop for progressive fusion in order to better adapt to the complexity of the CNN. Learning rates started with 0.00001 for baseline models and learning rate schedulers for progressive fusion, including reducing the learning rate by 0.1 every 5 epochs and a cyclical learning rate that oscillated every 5 epochs. Each model was trained on 20 epochs, though early stopping was introduced to progressive fusion when accuracy and loss plateaued in improvement. Performance was evaluated using validation accuracy, validation loss, and Area Under the Curve (AUC).

Results

Preliminary/Baseline Results

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

Two models were used as benchmarks with which to compare the progressive fusion model. Both the early and late fusion benchmark models were trained using the Adam optimizer, 20 epochs, and a batch size of 32 with a ResNet50 backbone model. While early fusion showed a slight advantage over late fusion in both validation accuracy and loss, both models performed quite poorly. A loss of around 0.693 is typical for binary classification with poor predictions (close to random guessing). This indicates that neither the early or late fusion model is effectively learning the relationship between input features (images and digital attributes) and labels. The accuracy of around 50% indicates that neither models are performing better than random guessing (50% for binary classification), reflecting an issue with the data, models, or fusion approach. In general, both fusion strategies show a limited ability to leverage cross-modal information effectively, which indicates that the current dataset itself may itself lack strong signal in either modality or require more complex interactions than just simple concatenation. When examining the class distribution for each label, class imbalance was eliminated as a possible source of poor performance, as both training and validation class distributions were almost exactly even. This leaves the following explanations for poor performance: the current model architecture may not be effective at extracting meaningful patterns, the models might be too simple or too complex for the given dataset, or metadata features might be noisy or weakly correlated with the labels. In order to eliminate issues with lack of complexity with model architecture, a baseline progressive fusion model was proposed.

Unlike the benchmark models, the progressive fusion model followed the encoder-decoder architecture described above, also trained in 20 epochs with a batch size of 32. The progressive fusion model only slightly outperformed the late fusion model in terms of accuracy, though early fusion still barely outperformed all other strategies in both validation accuracy and loss. Considering the benchmark models did not have the complex encoder-decoder architecture to leverage, it was still possible that the progressive fusion model could be optimized to improve performance and generalization using hyperparameter tuning.

Final Results

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

The first tuning strategy employed was to reduce the learning rate to 0.0001, with the goal of allowing the model to learn more effectively. Eventually, a learning rate scheduler was implemented based on its success in Huang et. al. (2023) and the recommendation of Sellat et. al. (2022). Two learning rate schedulers were explored: one where learning rate was reduced by 0.1 every five epochs and a cyclic learning rate, oscillating every five epochs. The reduction ratio was also reduced from 16 to 8 to preserve more information in the SE block bottleneck and enhance feature recalibration for progressive fusion tasks, though this parameter was not as precisely tuned as suggested by Hu et al. (2017). In order to address overfitting and improve generalization, an L2 regularizer was applied to metadata and dropout regularization layers were added, with multiple dropout values between 0.5 and 0.7 attempted. Moreover, the optimizer was changed to RMSProp due to its efficacy with CNNs and the vanishing gradient problem (Kashyap, 2024). Unfortunately, all of these strategies lead to severe overfitting, with a training

accuracy in the 70-80% range and a validation accuracy stagnating around 49-50%. The number of filters was also reduced in order to reduce overfitting while the batch size was reduced to 16, leading to a model with less severe—yet still present—overfitting. The final tuned progressive fusion model only outperformed the progressive fusion and late fusion baselines, but only slightly. Moreover, this model suffered from unusually high loss, likely as a result of overregularization.

As a "hail mary", a shallower backbone was attempted to see if a reduction in complexity would benefit performance and generalizability. Using the same SE block with a reduction ratio of 8, the shallower model used MobileNetV2 as a shallower encoder, with L2 regularization, constrained weights, a dropout of 0.5, the cyclical learning rate scheduler, 20 epochs, and a batch size of 32. The resulting validation accuracy was very slightly better than that of the early fusion model, though the early fusion model still outperformed the other models assessed in this paper in both validation accuracy and loss. With these results in mind, we conclude that progressive fusion did not show significant improvement over early and latest fusion strategies in any performance evaluation metric.

Discussion

Analysis of Results

There are a number of reasons behind the poor performance of the progressive fusion models, but the most likely is that the metadata features may not have been sufficiently informative to boost performance, which is crucial in multimodal fusion tasks. In general, if the metadata and the features derived from it lack strong correlation with the target variable, the fusions strategies—regardless of their complexity—will struggle to improve performance. Metadata should add complementary information that images alone provide. However, if the metadata is noisy or weakly correlated with the target variables, it becomes more of a distraction than a helpful input into the encoder. This is not surprising, as the results of the canonical correlation analysis (CCA) performed during data preprocessing found that the digital attributes such as brightness, spectral band values, and color magnitude were not strongly correlated with the target variables of snow_ice and leave_on_trees. As a result, the fusion mechanism likely could not effectively leverage the metadata.

Moreover, a major limitation mentioned by Huang et. al. (2023) is that the progressive fusion model is so complex that it requires meticulously tuned model weights, rendering pre-loaded weights useless. Due to the time and resource constraints on this project, I did not have either the time or appropriate resources to single handedly use weights trained exclusively from scratch. While weights used in the SE blocks, fusion layers, metadata processing layers, and fully connected layers were trained from scratch, while the ResNet50 and MobileNetV2 backbones used pre-trained weights. The SE blocks also did not have finely tuned reduction ratios as recommended by Hu et. al. (2017). This limitation was symptomatic of a larger one: the interaction between computing resources. Although the Apple M3 Max is considered a top-of-the-line model and Apple's most powerful machine, a high performance computing tool

like NVIDIA likely would have significantly reduced training time, thus allowing for far more time spent finely tuning the weights and other model parameters. It is also important to remember that the contents of this paper were done over the course of a four month semester. Regardless of how little time could have theoretically been spent training, working on this particular project for a year would have allowed much more time dedicated to fine tuning. This relationship between time and available resources also prevented experimentation with various epochs, batch sizes, and regularization methods.

Future Recommendations

As detailed in the analysis of results, a major way to potentially improve upon the research described in this paper would first be to set a much larger time frame for model training and tuning, as well as utilizing multiple researchers instead of just one. Regardless of the computational resources available, a larger team of researchers and a wider time frame would allow for sufficient time to be spent experimenting with hyperparameters such as learning rate, batch size, number of epochs, optimizers, regularization strategies, reduction ratios, and other architectures. It would also allow for sufficient time spent on training all model weights from scratch and the enormous amount of time necessary to finely tune said weights. Aside from an overall project structure change, many possible improvements can be made on a metadata-level.

One possible strategy could be to improve metadata quality with new features by exploring additional digital attributes that might correlate more strongly with the surface conditions, such as textural metrics like entropy or edge density (Haralick et. al., 1973). One could also incorporate environmental factors like soil moisture, temperature, and humidity (Kerr et. al., 2010) or NWDI, the normalized difference water index for remote sensing of vegetation liquid water from space (Gao, 1996). Feature engineering could also be used, where transformations are applied to derive richer metadata futures (Domingos, 2012). Dimensionality reduction using PCA or autoencoders could also be used to compress metadata into a more informative latent space (Hinton & Salakhutdinov, 2006), or even custom metrics tailored to specific conditions (Huete, 1988). One could also simplify fusion tasks by testing fusion strategies on simpler, synthetic tasks where the correlation between metadata and labels is stronger, which would allow the isolation and evaluation of the effectiveness of the fusion mechanisms themselves (Baltrusaitis et al., 2019). Finally, one could use attention mechanisms to dynamically weight the contribution of metadata during fusion, helping the model ignore irrelevant metadata when it is not helpful (Vaswani et. al., 2017).

Conclusion

This study evaluated multiple fusion strategies—early, late, and progressive—within an encoder-decoder architecture with SE blocks to predict surface conditions from multimodal data. The results indicated that early fusion slightly outperformed late and progressive fusion strategies, achieving the highest validation accuracy. Despite the theoretical advantages of progressive fusion, its implementation did not yield significant improvements, likely due to

metadata quality and the limitations of pretrained weights. Several factors contributed to these outcomes. The weak correlation between digital attributes and surface conditions, as evidenced by canonical correlation analysis, reduced the effectiveness of multimodal fusion. Additionally, the complexity of progressive fusion requires finely tuned model weights, which was constrained by computational resources and time. Regularization techniques, such as dropout and L2 penalties, mitigated overfitting but could not substantially improve validation performance.Future work should focus on enhancing metadata quality through new features (e.g., textural metrics, environmental data) and applying advanced feature engineering techniques. Simplifying fusion tasks or incorporating attention mechanisms to dynamically weight metadata contributions may also improve performance. Addressing these challenges with extended training time, additional resources, and collaboration with domain experts could unlock the full potential of progressive fusion for land-cover classification.

References

Alharbi, R., Alhichri, H., Ouni, R., Bazi, Y., & Alsabaan, M. (2023). Improving remote sensing scene classification using quality-based data augmentation. *International Journal of Remote Sensing*, *44*(6), 1749–1765. https://doi.org/10.1080/01431161.2023.2184213

Atef, I., Ahmed, W., Abdel-Maguid, R. H., Baraka, M., Darwish, W., & Senousi, A. M. (2023). Land use and land cover simulation based on integration of artificial neural networks with cellular automata-markov chain models applied to El-Fayoum governorate. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *X-1/W1-2023*, 771–777. https://doi.org/10.5194/isprs-annals-x-1-w1-2023-771-2023

Baltrusaitis, T., Ahuja, C., & Morency, L.-P. (2019). Multimodal Machine Learning: A Survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(2), 423–443. https://doi.org/10.1109/tpami.2018.2798607

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., & Torr, P. H. (2016). Fully-convolutional siamese networks for Object Tracking. *Lecture Notes in Computer Science*, 850–865. https://doi.org/10.1007/978-3-319-48881-3_56

Brabandere, B. D. (2024, May 22). *Late vs early sensor fusion for autonomous driving*. Segments.ai. https://segments.ai/blog/late-vs-early-sensor-fusion-a-comparison/

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834–848. https://doi.org/10.1109/tpami.2017.2699184

Chen, Z., & Cowan, N. (2013). Working memory inefficiency: Minimal information is utilized in visual recognition tasks. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *39*(5), 1449–1462. https://doi.org/10.1037/a0031790

DiCarlo, J. J., Zoccolan, D., & Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron*, *73*(3), 415–434. https://doi.org/10.1016/j.neuron.2012.01.010

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78–87. https://doi.org/10.1145/2347736.2347755

Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(2), 295–307. https://doi.org/10.1109/tpami.2015.2439281

Gao, B. (1996). Ndwi—a normalized difference water index for remote sensing of vegetation liquid water from space. *Remote Sensing of Environment*, *58*(3), 257–266. https://doi.org/10.1016/s0034-4257(96)00067-3

Global Learning and Observations to Benefit the Environment (GLOBE) Program, 29 September 2024, https://www.globe.gov/globe-data

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Chapter 10: Sequence Modeling: Recurrent and Recursive Nets. In *Deep Learning* (pp. 367–415). essay, MIT Press. Retrieved December 8, 2024, from https://www.deeplearningbook.org/contents/rnn.html.

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural Features for Image Classification. In *IEEE Transactions on Systems, Man, and Cybernetics* (Vol. 6, pp. 610–621). essay, Institute of Electrical and Electronics Engineers Inc. Retrieved December 8, 2024, from https://haralick.org/book_chapters/TexturalFeatures.pdf.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2016.90

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with Neural Networks. *Science*, *313*(5786), 504–507. https://doi.org/10.1126/science.1127647

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2018.00745

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. *International Journal of Applied Earth Observation and Geoinformation*, *122*, 103382. https://doi.org/10.1016/j.jag.2023.103382

Huete, A. R. (1988). A soil-adjusted vegetation index (SAVI). *Remote Sensing of Environment*, *25*(3), 295–309. https://doi.org/10.1016/0034-4257(88)90106-x

Jadon, S. (2020). A survey of loss functions for semantic segmentation. *IEE*. https://doi.org/10.48550/arXiv.2006.14822

Kashyap, P. (2024, November 2). *Understanding RMSPROP: A simple guide to one of Deep Learning's powerful optimizers*. Medium. https://medium.com/@piyushkashyap045/understanding-rmsprop-a-simple-guide-to-one-of-deep-learnings-powerful-optimizers-403baeed9922#:~:text=Using%20RMSProp%2C%20the%20moving%20average,slowdown%2C%20ultimately%20achieving%20better%20performance.

Kerr, Y. H., Waldteufel, P., Wigneron, J.-P., Delwart, S., Cabot, F., Boutin, J., Escorihuela, M.-J., Font, J., Reul, N., Gruhier, C., Juglea, S. E., Drinkwater, M. R., Hahne, A., Martín-Neira, M., & Mecklenburg, S. (2010). The smos mission: New tool for monitoring key elements ofthe global water cycle. *Proceedings of the IEEE*, *98*(5), 666–687. https://doi.org/10.1109/jproc.2010.2043032

Li, X., Zhang, G., Cui, H., Hou, S., Chen, Y., Li, Z., Li, H., & Wang, H. (2022). Progressive fusion learning: A Multimodal Joint Segmentation Framework for building extraction from optical and SAR images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *195*, 178–191. https://doi.org/10.1016/j.isprsjprs.2022.11.015

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2017.106

Low, R. D., Nelson, P. V., Soeffing, C., & Clark, A. (2021). Adopt a pixel 3 km: A multiscale data set linking remotely sensed land cover imagery with field based citizen science observation. *Frontiers in Climate*, *3*. https://doi.org/10.3389/fclim.2021.658063

Lv, H., Zhu, H., Zhu, R., Wu, F., Wang, C., Cai, M., & Zhang, K. (2024). Direction-guided Multiscale Feature Fusion Network for Geo-localization. *IEEE Transactions on Geoscience and Remote Sensing*, *62*, 1–13. https://doi.org/10.1109/tgrs.2024.3396912

Pulapakura, R. (2024, March 6). *Multimodal models and Fusion - A Complete Guide*. Medium. https://medium.com/@raj.pulapakura/multimodal-models-and-fusion-a-complete-guide-225ca91f6861

Qin, R., & Liu, T. (2022). A review of Landcover classification with very-high resolution remotely sensed optical images—analysis unit, model scalability and transferability. *Remote Sensing*, *14*(3), 646. https://doi.org/10.3390/rs14030646

Rozental, A., & Biton, D. (2019). *Amobee at SemEval-2019 Tasks 5 and 6: Multiple Choice CNN Over Contextual Embedding*. https://doi.org/ https://doi.org/10.48550/arXiv.1904.08292

Sellat, Q., Bisoy, S. K., & Priyadarshini, R. (2022). Semantic segmentation for self-driving cars using Deep Learning. Cognitive Big Data Intelligence with a Metaheuristic Approach, 211–238. https://doi.org/10.1016/b978-0-323-85117-6.00002-9

Shankar, S., Thompson, L., & Fitreau, M. (n.d.). *Progressive Fusion for Multimodal Integration*. https://doi.org/https://doi.org/10.48550/arXiv.2209.00302

Sun, H., Xie, Y., Ren, D., Wen, F., Tong, L., & Chang, L. (2024). MMLN: Multi-directional and Multi-constraint Learning Network for remote sensing imagery semantic segmentation. *IEEE*

Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 1–16. https://doi.org/10.1109/jstars.2024.3403854

Talmi, I., Mechrez, R., & Zelnik-Manor, L. (2017). Template matching with deformable diversity similarity. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1311–1319. https://doi.org/10.1109/cvpr.2017.144

Tsotsos, J. K., Rodríguez-Sánchez, A. J., Rothenstein, A. L., & Simine, E. (2008). The different stages of visual recognition need different attentional binding strategies. *Brain Research*, *1225*, 119–132. https://doi.org/10.1016/j.brainres.2008.05.038

Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. Neural Information Processing Systems.

Xiao, J., Zhang, D., Li, J., & Liu, J. (2024). A study on the classification of complexly shaped cultivated land considering multi-scale features and edge priors. *Environmental Monitoring and Assessment*, *196*(9). https://doi.org/10.1007/s10661-024-12966-8

Yuan, H., Zhang, Z., Rong, X., Feng, D., Zhang, S., & Yang, S. (2023). MPFFNet: LULC classification model for high-resolution remote sensing images with multi-path feature fusion. *International Journal of Remote Sensing*, *44*(19), 6089–6116. https://doi.org/10.1080/01431161.2023.2261153