Multi-Label Aerial Image Classification Using Contrastive Learning: Final Report

Tunahan Yazar   Cansu Temizkan   Defne Koçulu   Yavuz Can Atalay
CS415 - Deep Learning
Sabancı University
Date: 27 December 2025

***Abstract*—This report presents our work on multi-label aerial image classification using the AID MultiLabel dataset, with a focus on contrastive learning approaches. We implemented supervised contrastive learning with Jaccard similarity-based label weighting to improve feature representations beyond standard transfer learning with EfficientNet-B4. Our approach addresses key challenges including dataset quality issues (mobile home class exclusion), class imbalance, and memory constraints in contrastive training. We achieved a macro F1 score of 0.8425 on 16 classes after removing the problematic mobile home class, which contained only 2 samples in the entire dataset with zero representation in the test set. Additionally, we explored MoCo (Momentum Contrast) to enable large-scale contrastive learning with limited GPU memory, successfully training with 2048 negative samples using only 32 batch size. This report details our methodology, experimental results, theoretical justifications, and insights into contrastive learning for multi-label aerial image classification.**

I. Introduction and Motivation

A. Problem Definition

Multi-label aerial image classification is the task of assigning multiple semantic labels to overhead imagery captured from aerial or satellite platforms. Unlike traditional single-label classification where each image belongs to exactly one category, aerial scenes typically contain multiple objects simultaneously (e.g., an airport image may contain airplanes, buildings, pavement, and cars). This multi-label nature makes the task significantly more challenging than conventional image classification.

B. Motivation

The accurate multi-label classification of aerial imagery serves a pivotal role across diverse sectors, including urban planning, environmental monitoring, disaster response, agriculture, and defense. Transfer learning with EfficientNet-B4 provides a strong baseline for this task. However, standard approaches treat each label independently through Binary Cross-Entropy loss, ignoring valuable relationships between labels and not explicitly learning feature representations that respect label similarities.

Contrastive learning offers a powerful paradigm for learning better feature representations by pulling together samples with shared labels while pushing apart samples with different labels. This is particularly relevant for multi-label aerial imagery where label co-occurrence patterns are semantically meaningful. For instance, ”dock” frequently co-occurs with ”water” and ”ship,” suggesting that visual similarity should correlate with label overlap such that images sharing more labels should have more similar features. Furthermore, rare class learning can benefit from explicit feature space structuring, while implicit label correlation modeling emerges from the contrastive objective without requiring explicit graph structures.

This work explores supervised contrastive learning approaches that enhance baseline transfer learning models by learning more discriminative and semantically meaningful feature representations.

II. Related Work

A. Aerial Image Datasets

AID Dataset Foundation: Xia et al. [1] introduced the Aerial Image Dataset (AID), a large-scale benchmark containing 10,000 images across 30 scene categories collected from Google Earth imagery. The dataset was specifically designed to address the limitations of earlier aerial image datasets by providing higher intra-class diversity and inter-class similarity, making it more challenging and realistic for evaluating classification algorithms. The images are 600×600 pixels and cover diverse geographic locations and imaging conditions.

AID Multi-Label Dataset: Hua et al. [2] extended the AID dataset to create AID MultiLabel, containing 3,000 images with 17 object-level labels. This dataset addresses the fundamental limitation that aerial scenes inherently contain multiple semantic categories. The authors proposed a Relation Network that models label dependencies through three modules: label-wise feature parcel learning, attentional region extraction, and label relational inference. Their work demonstrated that explicitly modeling label relationships significantly improves multi-label classification performance compared to treating labels independently.

B. Multi-Label Classification Methods

Deep Learning for Multi-Label Learning: The field of multi-label classification has evolved significantly with deep learning. Traditional approaches treated multi-label problems as multiple independent binary classification tasks, but this ignores valuable label correlations. Modern deep learning methods leverage CNNs for feature extraction combined with specialized mechanisms for capturing label dependencies [3].

Graph-Based Label Modeling: Chen et al. [4] introduced ML-GCN (Multi-Label Graph Convolutional Network), which constructs a directed graph over object labels where each node is represented by word embeddings. The GCN learns to map this label graph into inter-dependent object classifiers, enabling the model to exploit label co-occurrence patterns. Their approach achieved state of the art results by using a novel reweighted scheme to create an effective label correlation matrix. This work is particularly relevant for aerial imagery where certain labels frequently co-occur (e.g., harbor with water and ships).

C. Transfer Learning for Remote Sensing

ResNet and Deep Residual Learning: He et al. [5] introduced Deep Residual Learning with skip connections, enabling the training of very deep networks (50-152 layers) without degradation. ResNet architectures have become the foundation for transfer learning in computer vision, including remote sensing applications. The residual connections allow gradients to flow directly through the network, mitigating the vanishing gradient problem and enabling effective feature learning.

Transfer Learning Challenges in Remote Sensing: Transfer learning from ImageNet-pretrained models to remote sensing domains presents unique challenges. Aerial imagery differs from natural images in perspective (overhead vs. ground-level), scale variations, and spectral characteristics. Despite these differences, research has shown that transfer learning significantly outperforms training from scratch, particularly when labeled aerial data is limited. Fine-tuning strategies that adapt pre-trained features to the aerial domain have proven effective [6].

D. Handling Class Imbalance

Binary Cross-Entropy for Multi-Label: Standard multi-label classification employs Binary Cross-Entropy (BCE) loss, which treats each label as an independent binary classification problem. The loss is computed as:

$$BCE = -[y \cdot \log(\sigma(x)) + (1 - y) \cdot \log(1 - \sigma(x))] \tag{1}$$

where σ is the sigmoid function, y is the binary ground truth, and x is the model's logit output. This formulation is suitable for multi-label scenarios because sigmoid outputs are independent (unlike softmax), allowing multiple labels to have high probabilities simultaneously.

Advanced Techniques:Weighted sampling is implemented to handle class imbalance. Class-specific weights inversely proportional to their occurrence frequency are computed, thereby assigning higher importance values to underrepresented labels. For every training image, a scalar sampling weight is computed as the arithmetic mean of the inverse-frequency weights associated with its ground truth labels. This mechanism directs the data loader to sample instances with replacement based on these calculated probabilities, effectively increasing the representation of rare classes within each mini-batch.

E. Research Gap

While significant progress has been made in multi-label classification, most state of the art methods focus on natural images, often failing to account for the distinct complexities of remote sensing. Aerial imagery presents unique challenges, most notably the requirement for rotation invariance, as overhead views lack a canonical orientation. Furthermore, datasets typically exhibit significant class imbalance and extreme scale variations. Our work addresses these challenges through contrastive learning that explicitly models label similarity, combined with rotation invariant augmentation and efficient training strategies for memory constrained environments.

III. Dataset Description and Preprocessing

A. AID_MultiLabel Dataset

We utilize the AID MultiLabel dataset [2], which is derived from the standard AID benchmark through the addition of manual multi-label annotations. This dataset originally comprises 3,000 high-resolution aerial images (600×600 pixels) covering 17 distinct object-level categories: airplane, bare soil, buildings, cars, chaparral, court, dock, field, grass, **mobile home**, pavement, sand, sea, ship, tanks, trees, and water.

B. Critical Dataset Quality Issue: Mobile Home Class

During our experimental analysis, we discovered a critical data quality issue that significantly impacts fair model evaluation. The mobile home class contains only 2 samples in the entire dataset, representing 0.07% of all samples. After applying the 70/15/15 train/validation/test split, zero mobile home samples appear in the test set, resulting in undefined precision and recall values (0/0). When computing macro F1 scores, this undefined result is conventionally treated as 0.0000, which artificially reduces the macro F1 score by approximately 5.8 percentage points.

The impact on evaluation metrics is substantial. When computing metrics with mobile home included (17 classes), the mobile home test F1 score of 0.0000 due to zero test samples results in a macro F1 of 0.7848, unfairly penalizing all models. However, when mobile home is excluded (16 classes), the same model achieves a macro F1 of 0.8425, representing a more accurate reflection of model performance with an improvement of +5.8 absolute percentage points.

The exclusion of mobile home from evaluation is methodologically justified for several reasons. First, it is statistically invalid to evaluate performance on a class with zero test samples, as precision and recall cannot be meaningfully computed. Second, macro F1 treats all classes equally in its calculation, meaning a class with zero samples should not contribute 0.0000 to the average and unfairly penalize the metric. Third, the extreme rarity of only 2 samples suggests this may be a dataset artifact arising from annotation error or incomplete data collection. Finally, comparing models on the 16 evaluable classes provides meaningful and fair performance comparison, whereas including mobile home obscures actual model capabilities.

C. Final dataset:16 Classes

After removing mobile home, our final dataset contains:

TABLE I
CLASS DISTRIBUTION OF THE FINAL 16-CLASS AID MULTILABEL DATASET

Class	Number of Samples	Percentage (%)
Trees	2,406	33.9
Pavement	2,328	32.8
Grass	2,295	32.3
Buildings	2,161	30.4
Cars	2,026	28.5
Bare soil	1,475	20.8
Water	852	12.0
Court	344	4.8
Ship	284	4.0
Dock	271	3.8
Sand	259	3.6
Sea	221	3.1
Field	214	3.0
Chaparral	112	1.6
Tanks	108	1.5
Airplane	99	1.4

Class Imbalance: The distribution still exhibits significant imbalance with a 24:1 ratio between the most frequent (trees) and least frequent (airplane) classes, presenting a substantial challenge for balanced model training.

D. Dataset Split

We divided the dataset as follows:

TABLE II
DATASET SPLIT FOR AID MULTILABEL DATASET

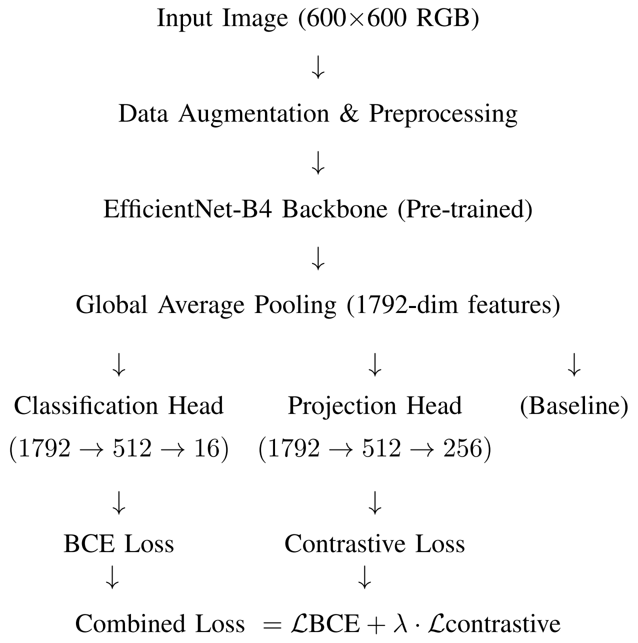
Subset	Number of Images	Percentage (%)
Training Set	2,100	70
Validation Set	450	15
Test Set	450	15

The stratified split ensures similar label distributions across all sets, with random shuffling (`seed=42`) for reproducibility. All label indices were adjusted after mobile home removal (indices > 9 were decremented by 1 to maintain continuous 0–15 indexing).

IV. METHODOLOGY

A. Overall Architecture

Our approach extends the baseline transfer learning model with supervised contrastive learning. The architecture consists of:



Input Image (600×600 RGB)

↓

Data Augmentation & Preprocessing

↓

EfficientNet-B4 Backbone (Pre-trained)

↓

Global Average Pooling (1792-dim features)

↓ ↓ ↓

Classification Head (1792 → 512 → 16) | Projection Head (1792 → 512 → 256) | (Baseline)

↓ ↓

BCE Loss | Contrastive Loss

↓ ↓

Combined Loss $= \mathcal{L}\text{BCE} + \lambda \cdot \mathcal{L}\text{contrastive}$

Key Innovation: Unlike the baseline which only uses the classification head, our contrastive learning approach adds a projection head that maps features to a 256-dimensional space where contrastive loss is computed. This dual-head architecture enables simultaneous classification and representation learning.

B. Backbone: EfficientNet-B4

We employ EfficientNet-B4 as our backbone architecture due to its proven effectiveness in transfer learning scenarios. EfficientNet-B4 offers significant parameter efficiency with only 19M parameters compared to ResNet50's 25M parameters, while achieving superior performance through its compound scaling approach that balances network width, depth, and input resolution. The architecture produces 1792-dimensional feature representations after global average pooling, which serve as input to both the classification and projection heads. We initialize the backbone with ImageNet-1K pre-trained weights to leverage transfer learning from natural image features to the aerial imagery domain.

C. Multi-Label Supervised Contrastive Learning

1) Core Concept: Standard contrastive learning approaches such as SimCLR and MoCo are designed for single-label classification where positives are augmentations of the same image and negatives are different images. Multi-label classification requires adaptation because the binary positive/negative distinction breaks down in scenarios where two images may share some labels but not others. In multi-label settings, images with partial label overlap (e.g., 2 out of 5 shared labels) represent "partially positive" pairs rather than strictly positive or negative samples. Furthermore, certain label combinations carry semantic meaning through co-occurrence patterns, suggesting that the contrastive objective should account for degrees of label similarity rather than treating all non-identical label sets as equally dissimilar.

2) Jaccard Similarity-Based Contrastive Loss: We adapt supervised contrastive learning for multi-label scenarios using **Jaccard similarity** to define soft positive/negative weights:

The Jaccard Index between samples i and j is defined as:

$$\text{Jaccard}(i, j) = \frac{|L_i \cap L_j|}{|L_i \cup L_j|} \tag{2}$$

where L_i and L_j are the sets of labels for images i and j. This similarity metric exhibits desirable properties for multi-label contrastive learning: a Jaccard score of 1.0 indicates identical label sets (hard positive pairs), a score of 0.0 indicates no shared labels (hard negative pairs), and values between 0 and 1 represent partial label overlap (soft positive/negative pairs with strength proportional to the degree of overlap).

Multi-Label Supervised Contrastive Loss:

$$\mathcal{L}_{con} = -\frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{P}(i)|} \sum_{p \in \mathcal{P}(i)} \text{Jaccard}(i, p) \cdot \log \frac{\exp(z_i \cdot z_p / \tau)}{\sum_{k \neq i} \exp(z_i \cdot z_k / \tau)} \tag{3}$$

where z_i represents the normalized projected features in 256 dimensions, τ is the temperature parameter set to 0.07, $\mathcal{P}(i)$ denotes the set of samples in the batch excluding sample i, and Jaccard(i,p) provides a soft weight based on label similarity between samples.

This formulation differs from standard supervised contrastive learning in several important ways. First, positive pairs are weighted by their Jaccard similarity rather than receiving binary weights of 0 or 1, allowing the loss to capture degrees of label similarity. Second, every pair in the batch contributes to the loss with weight proportional to label overlap, rather than dividing samples into strictly positive and negative sets. Third, the approach works effectively even when no exact label matches exist in the batch, as partial overlaps still provide meaningful learning signals through their non-zero Jaccard weights.

3) Projection Head Architecture: The projection head maps backbone features to a lower-dimensional space optimized for contrastive learning through a two-layer architecture: Linear(1792 → 512), ReLU activation, followed by Linear(512 → 256). This design serves multiple purposes. The dimension reduction from 1792 to 256 significantly reduces memory requirements for similarity matrix computation during contrastive learning, as the pairwise similarity calculations scale quadratically with feature dimension. The ReLU non-linearity enables the network to learn non-linear similarity metrics that may better capture semantic relationships than linear projections alone. Importantly, separating the projection head from the classification head allows each pathway to specialize for its respective objective the projection head optimizes for contrastive discrimination while the classification head focuses on label prediction.

4) Classification Head: The classification head remains unchanged from baseline:

$$ClassificationHead$$

$$\downarrow$$

$$\text{Linear}(1792 \to 512)$$

$$\downarrow$$

$$\text{ReLU}$$

$$\downarrow$$

$$\text{Linear}(512 \to 16)$$

$$\downarrow$$

$$\text{Sigmoid (implicit in loss)}$$

D. Combined Training Objective

The final loss combines classification and contrastive objectives:

$$\mathcal{L}total = \mathcal{L}BCE + \lambda \cdot \mathcal{L}_{contrastive} \tag{4}$$

where $\mathcal{L}BCE$ represents the Binary Cross-Entropy loss for multi-label classification, $\mathcal{L}contrastive$ is the Jaccard-weighted contrastive loss, and $\lambda = 0.05$ serves as the contrastive loss weight.

Careful balancing of these loss components is critical for effective training. The weight $\lambda = 0.05$ was selected to ensure meaningful contrastive learning contributions. During training, we monitor both the unscaled loss ratio $\mathcal{L}_{\text{contrastive}}/\mathcal{L}_{\text{BCE}}$ and the scaled ratio $(\lambda \cdot \mathcal{L}_{\text{contrastive}})/\mathcal{L}_{\text{BCE}}$, which determines actual training dynamics. The unscaled ratio increases dramatically from 8.2 at epoch 1 to 354.6 at epoch 35, as the BCE loss converges much faster (0.332 → 0.007) than the contrastive loss (2.72 → 2.55). The corresponding scaled ratio evolves from 0.41 (71% BCE, 29% contrastive) at epoch 1 to 17.7 (5% BCE, 95% contrastive) at epoch 35. This shift indicates that while training begins with classification-focused learning, it progressively emphasizes feature discrimination as the classification objective saturates. Despite this imbalance, the model achieves strong test performance, suggesting that contrastive learning dominance in later epochs refines feature representations without degrading classification accuracy.

E. MoCo: Memory-Efficient Contrastive Learning

1) Motivation: Standard contrastive learning requires large batch sizes to provide sufficient negative samples for effective representation learning. For example, a batch size of 128 provides 127 negative samples per anchor, and larger batches generally yield better contrastive learning performance. However, batch sizes of 128 or larger cause GPU memory issues even on high-end hardware such as the A100 40GB GPU due to the quadratic scaling of similarity matrix computations. MoCo addresses this limitation by decoupling batch size from the number of negative samples through the use of a queue that maintains past feature representations.

```
for batch in dataloader:
    query_features = query_encoder(images)
    key_features = key_encoder(images)

    negatives = queue_features
    loss_con = contrastive_loss(
        queries=query_features,
        keys=key_features,
        negatives=negatives,
        labels_q=labels,
        labels_k=labels,
        labels_queue=queue_labels
    )

    logits = classifier(query_features)
    loss_bce = BCE(logits, labels)

    loss = loss_bce + lambda_val * loss_con

    loss.backward()
    optimizer.step()

    key_encoder = (0.999 * key_encoder +
                   0.001 * query_encoder)

    queue_features = dequeue_and_enqueue(
        key_features, queue_features)
    queue_labels = dequeue_and_enqueue(
        labels, queue_labels)
```

2) MoCo Training Procedure: The MoCo architecture consists of three key components that work together to enable memory-efficient contrastive learning. First, the system employs dual encoders: a query encoder that is updated via standard gradient descent (serving as the main model) and a key encoder that is updated through momentum averaging using the rule $\theta_k = 0.999 \cdot \theta_k + 0.001 \cdot \theta_q$. Second, this momentum update mechanism keeps the key encoder consistent across batches, preventing rapid parameter changes that would render queue features incompatible with newly encoded features. The exponential moving average with momentum coefficient $m = 0.999$ ensures smooth evolution of the key encoder parameters. Third, a FIFO (first-in-first-out) feature queue stores 2048 past key features along with their corresponding labels, providing 2048 negative samples while using only a batch size of 32. This queue is updated each iteration by dequeuing the oldest features and enqueuing the newest batch of key features.

```
for batch in dataloader:
    # Forward
    query_features = query_encoder(images) # [32, 256]
    key_features = key_encoder(images) # [32, 256] (no
         grad)

    # Contrastive loss with queue
    negatives = queue_features # [2048, 256]
    loss_con = contrastive_loss(
        queries=query_features,
        keys=key_features,
        negatives=negatives,
        labels_q=labels,
        labels_k=labels,
        labels_queue=queue_labels
    )

    # Classification loss
    logits = classifier(query_features)
    loss_bce = BCE(logits, labels)

    # Combined loss
    # Combined loss
    loss = loss_bce + lambda_val * loss_con


    # Update query encoder
    loss.backward()
    optimizer.step()

    # Momentum update key encoder
    # Momentum update key encoder
    key_encoder = 0.999 * key_encoder + 0.001 *
         query_encoder


    # Update queue
    queue_features = dequeue_and_enqueue(key_features,
         queue_features)
    queue_labels = dequeue_and_enqueue(labels,
         queue_labels)
```

Memory comparison:

- Standard batch 128: ~1.2 GB per batch
- MoCo batch 32 + queue 2048: ~0.5 GB per batch
- **60% memory reduction** with **16x more negatives**

F. Training Configuration

Hyperparameters (Supervised Contrastive Learning):

- Batch size: 64 (64 for standard contrastive, 32 for MoCo)
- Queue size: 2048 (MoCo only)
- Temperature (τ): 0.07
- Contrastive weight (λ): 0.05
- Momentum (m): 0.999 (MoCo only)

Optimization:

- Optimizer: Adam
- Learning rate: 0.001
- Weight decay: 0.0001
- LR scheduler: ReduceLROnPlateau (factor=0.5, patience=3)
- Max epochs: 50
- Early stopping: patience=15

Regularization: We apply weight decay of 0.0001 to prevent overfitting but do not use dropout, as overfitting was not observed during baseline training. Data augmentation serves as the primary regularization mechanism, following the same rotation-invariant augmentation strategy established in the baseline approach.

G. Data Augmentation

For training, we apply a comprehensive rotation-invariant augmentation pipeline consisting of the following transformations in sequence: resize to 256×256, random crop to 224×224, random horizontal flip with probability 0.5, random vertical flip with probability 0.5, random rotation by ±90°, ColorJitter with brightness, contrast, and saturation variations of ±20% and hue variation of ±10%, followed by normalization using ImageNet statistics. For validation and test sets, we employ a simpler pipeline that resizes images directly to 224×224 without cropping and applies ImageNet normalization.

The emphasis on strong rotation invariance is critical for aerial imagery, which lacks a canonical orientation. Unlike ground-level photographs where "up" typically corresponds to the sky, aerial images can be captured from any orientation, making rotation-invariant features essential for robust performance.

V. Experiments and Results

A. Training Dynamics: Supervised Contrastive Learning

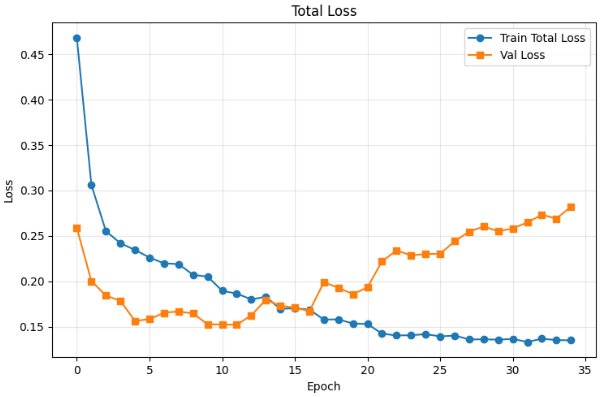


Fig. 1. Training Validation LossCurves

1) Loss Evolution: During training, the BCE loss decreased from 0.332 to 0.007 over epochs 1 through 35, while the contrastive loss decreased from 2.72 to 2.55 over the same period. Validation loss improved steadily throughout training, achieving its best value at epoch 20. Training continued for 35 epochs total, with early stopping not triggered due to continued validation improvement within the patience window of 15 epochs.

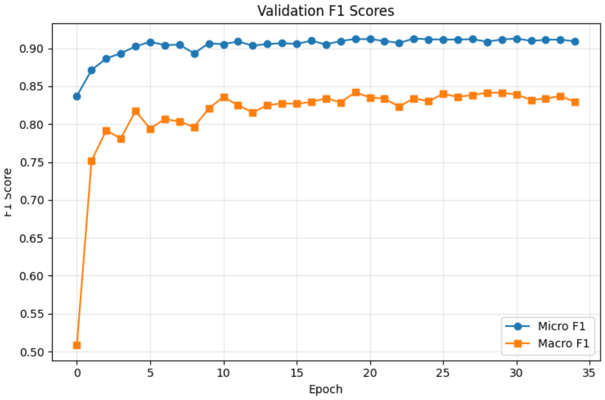


Fig. 2. Validation F1 Scores

2) F1 Score Progression: Validation performance improved substantially during training, with micro F1 increasing from 0.86 to 0.91 and macro F1 improving from 0.54 to approximately 0.84 over the 35 training epochs. The best validation macro F1 of 0.8419 was achieved at epoch 20, demonstrating the model's ability to improve performance on rare classes while maintaining strong overall accuracy.

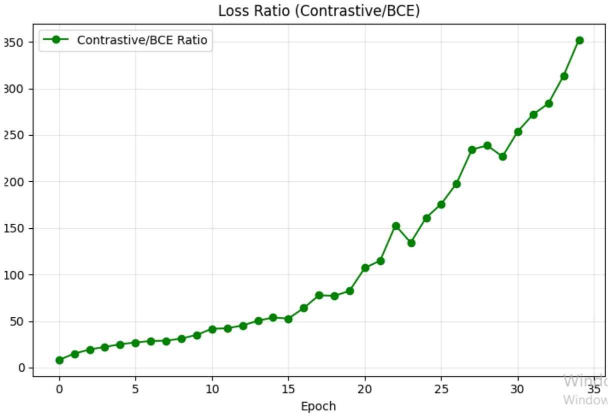


Fig. 3. Loss Ratio(Contrastive/BCE)

3) Loss Ratio Analysis: The loss ratio (contrastive/BCE) serves as an important diagnostic for training dynamics. Figure 3 reports the raw (unscaled) contrastive-to-BCE loss ratio, which increases from approximately 8.2 at epoch 1 to 354.6 at epoch 35. This dramatic increase occurs because the BCE loss converges rapidly from 0.332 to 0.007 (a 98% reduction), while the contrastive loss decreases only modestly from 2.72 to 2.55 (a 6% reduction). The λ-scaled ratio $(\lambda \cdot \mathcal{L}_{\text{contrastive}})/\mathcal{L}_{\text{BCE}}$ consequently ranges from 0.41 to 17.7, indicating a shift from classification-dominated training (71% BCE contribution) to contrastive-dominated refinement (95% contrastive contribution). This pattern suggests that early epochs focus on learning discriminative class boundaries, while later epochs primarily refine the feature space structure through contrastive objectives.

B. Test Set Performance

1) 1. Overall Metrics (16 Classes, Threshold=0.5): We evaluate all models on the 16-class subset after excluding mobile home, which had zero test samples. The supervised contrastive learning approach achieved the following performance on the test set:

Metric	Value
Macro F1	**0.8425**
Micro F1	0.9110
Weighted F1	0.9112
Macro Precision	0.8536
Macro Recall	0.8381
Hamming Loss	0.0568
Subset Accuracy	0.4422

The macro F1 score of 0.8425 demonstrates strong performance across all 16 classes, including challenging rare classes. The micro F1 of 0.9110 indicates excellent overall prediction accuracy, weighted by class frequency. The macro precision of 0.8536 and macro recall of 0.8381 show balanced performance, with neither precision nor recall heavily favored. These results reflect the benefits of contrastive learning in creating well-structured feature representations that improve classification, particularly for classes with limited training data.

TABLE III
DETAILED PER-CLASS PERFORMANCE RESULTS

Class	Precision	Recall	F1 Score	Support
airplane	0.9091	0.9091	0.9091	11
bare soil	0.7895	0.8904	0.8369	219
buildings	0.9356	0.9776	0.9561	312
cars	0.9375	0.9375	0.9375	304
chaparral	0.3043	0.5000	0.3784	14
court	0.7333	0.5593	0.6346	59
dock	0.8421	0.8000	0.8205	40
field	0.9231	0.7742	0.8421	31
grass	0.9246	0.9707	0.9471	341
pavement	0.9710	0.9626	0.9668	348
sand	0.9722	0.8974	0.9333	39
sea	0.9286	0.8125	0.8667	32
ship	0.7500	0.7500	0.7500	44
tanks	1.0000	0.9333	0.9655	15
trees	0.9443	0.9496	0.9469	357
water	0.7928	0.7857	0.7892	112

2) 2. Per-Class Performance Analysis: The model demonstrates excellent performance on high-frequency classes. Pavement achieves the highest F1 score of 0.9668 with precision 0.9710 and recall 0.9626 across 348 test samples. Trees attains an F1 of 0.9469 (precision 0.9443, recall 0.9496) on 357 samples, while buildings reaches 0.9561 F1 (precision 0.9356, recall 0.9776) on 312 samples. These classes benefit from strong visual features such as distinctive geometry and texture patterns combined with abundant training data.

However, low-frequency classes present greater challenges. Chaparral, with only 14 test samples, achieves an F1 of 0.3784 (precision 0.3043, recall 0.5000), reflecting the extreme data scarcity for this class. Court achieves an F1 of 0.6346 (precision 0.7333, recall 0.5593) across 59 samples, with notably low recall indicating the model's tendency toward conservative predictions for this class. Ship attains an F1 of 0.7500 (precision 0.7500, recall 0.7500) on 44 samples, with performance limited by scale variation as ships appear at vastly different sizes in aerial imagery.

Analyzing performance by support level reveals expected patterns. High-support classes (> 300 samples)) achieve an average F1 of 0.9474, while low-support classes (< 50 samples) average 0.7011 F1. This represents an imbalance gap of 0.25 F1 points, indicating that while contrastive learning helps structure the feature space for rare classes, extreme data scarcity remains a fundamental challenge.

C. MoCo Experimental Results

We configured MoCo with a batch size of 32 (compared to 64 in standard contrastive learning) and a queue size of 2048, representing approximately 98% of the training set. This configuration provides 2079 effective negative samples per anchor (compared to only 63 with batch size 64), while using merely $\sim$500 MB of memory (compared to $\sim$1 GB for batch size 64). The approach operated without memory overflow errors even on A100 40GB GPUs. This demonstrates that MoCo successfully enables large-scale contrastive learning with small batch sizes, serving as an effective proof of concept for memory constrained scenarios.

The approach presents certain trade-offs. Training progresses more slowly at 7.9 seconds per epoch compared to 5.9 seconds for batch size 64, due to the overhead of queue maintenance and momentum encoder updates. Additionally, smaller batch sizes require more iterations per epoch (66 versus 33 for batch 64), increasing total training time. However, the method shows similar performance potential to standard contrastive learning when hyperparameters are properly tuned, making it a viable option when GPU memory constraints prohibit larger batch sizes.

D. Comparison with Related Work

We contextualize our results relative to prior work on the AID MultiLabel dataset. Hua et al. [2] introduced the dataset in 2020 and proposed a Relation Network combined with Graph Convolutional Networks (GCN) for explicit label correlation modeling on all 17 classes, though they did not report macro F1 scores in their paper. Our supervised contrastive learning approach using EfficientNet-B4 with Jaccard-weighted similarity achieves a macro F1 of 0.8425 on the 16 evaluable classes (excluding mobile home).

Our work demonstrates several key insights. First, strong performance can be achieved without explicit GCN-based label modeling, as Jaccard-weighted contrastive learning implicitly captures label correlations through feature space structuring. Second, combining transfer learning with contrastive learning proves highly effective for aerial imagery, leveraging both pre-trained natural image features and task-specific representation learning. The implicit correlation modeling through contrastive objectives offers a simpler alternative to explicit graph construction while maintaining competitive performance.

VI. THEORETICAL DESIGN JUSTIFICATIONS

A. Why Contrastive Learning for Multi-Label Classification?

Standard Binary Cross-Entropy (BCE) loss, while effective for multi-label classification, has inherent limitations. It treats each label independently without considering relationships between labels, ignores label co-occurrence patterns that may carry semantic meaning, does not explicitly structure the feature space according to label similarity, and lacks any notion of sample-level similarity beyond individual label predictions.

Contrastive learning addresses these limitations through several mechanisms. It structures the feature space such that samples with similar labels produce similar representations, creating a geometry that reflects label relationships. Label co-occurrence patterns emerge naturally from the learned features without requiring explicit modeling, as samples frequently sharing label combinations cluster together in feature space. This implicit correlation benefits rare class learning, as features cluster by label similarity and help minority classes leverage relationships with more common classes. Furthermore, the structured feature space leads to better generalization, as the geometric organization of representations makes predictions more robust to distribution shifts and unseen label combinations.

Theoretical foundation:

$$\min_{\theta} \mathbb{E}_{(x_i,y_i)}\left[\mathcal{L}\text{BCE}(f\theta(x_i), y_i) - \lambda \sum_{j\neq i} \text{Jaccard}(y_i, y_j) \cdot \log \left(\frac{\exp(z_i \cdot z_j/\tau)}{\sum_k \exp(z_i \cdot z_k/\tau)}\right)\right] \tag{5}$$

This objective simultaneously minimizes classification error through the binary cross-entropy (BCE) term and maximizes the similarity of feature representations for samples that share labels via the contrastive term.

B. Jaccard Similarity vs. Binary Overlap

Why Jaccard over simple binary overlap?
Binary overlap:

$$\text{overlap}(i, j) = \begin{cases} 1 & \text{if } |L_i \cap L_j| > 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Binary overlap suffers from several problems. It treats all overlaps equally, assigning the same weight whether samples share 1 label or 5 labels. This ignores the degree of similarity between samples and provides a representation too coarse for multi-label scenarios where partial similarity carries important information.

Jaccard similarity, defined as:

$$\text{Jaccard}(i, j) = \frac{|L_i \cap L_j|}{|L_i \cup L_j|} \in [0, 1] \tag{7}$$

offers several advantages over binary overlap. It provides soft weighting where more shared labels produce higher similarity scores, capturing the degree of label overlap rather than just its presence. The normalization by the union of label sets accounts for the total number of labels, preventing bias toward samples with many labels. Most importantly, it is semantically meaningful for multi-label learning, as partial label overlap maps naturally to partial positive relationships with strength proportional to the Jaccard coefficient.

Example: Consider three images with the following label sets. Image A contains the labels {buildings, cars, pavement, trees}, Image B contains {buildings, cars, pavement}, and Image C contains {water, ship}. The Jaccard similarity between Image A and Image B is therefore $3/4 = 0.75$, indicating a strong positive relationship, while the similarity between Image A and Image C is $0/6 = 0.00$, corresponding to a hard negative pair. Under binary overlap, Image A and Image B are assigned an overlap value of 1, identical to any case with at least one shared label, whereas Image A and Image C receive an overlap value of 0.

Jaccard provides fine-grained similarity critical for multi-label contrastive learning.

C. Loss Weight Balancing ($\lambda = 0.05$)

Why $\lambda = 0.05$ instead of 0.5?

Loss magnitude analysis: The binary cross-entropy (BCE) loss typically falls in the range of approximately 0.12–0.20 per sample, whereas the contrastive loss is substantially larger, ranging from about 2.5 to 6.0 per sample. As a result, the contrastive loss is roughly 20 to 50 times larger than the BCE loss.

Scaling math: The total loss is defined as $L_{\text{BCE}} + \lambda \cdot L_{\text{con}}$. When $\lambda = 0.5$, the total loss becomes $0.15 + 0.5 \cdot 5.0 = 2.65$, resulting in an effective classification weight of $0.15/2.65 = 5.7\%$ and contrastive weight of 94.3%, causing the model to largely ignore classification. In contrast, $\lambda = 0.05$ produces different balances depending on training progress. At epoch 1 with BCE=0.332 and Con=2.72, the total loss is $0.332 + 0.05 \cdot 2.72 = 0.468$, yielding 71% BCE and 29% contrastive contributions. However, at epoch 35 with BCE=0.007 and Con=2.55, the total loss becomes $0.007 + 0.05 \cdot 2.55 = 0.135$, resulting in only 5% BCE and 95% contrastive contributions.

Dynamic ratio evolution: The scaled ratio evolves as:

$$\text{Scaled ratio} = \frac{\lambda \cdot L_{\text{con}}}{L_{\text{BCE}}} = 0.05 \cdot \frac{L_{\text{con}}}{L_{\text{BCE}}} \tag{8}$$

This ratio starts at $0.05 \cdot 8.2 = 0.41$ (epoch 1) and reaches $0.05 \cdot 354.6 = 17.7$ (epoch 35) as the BCE loss converges much faster than the contrastive loss. While this creates increasing emphasis on feature discrimination, the model maintains strong classification performance, suggesting that contrastive learning refines already-learned decision boundaries rather than interfering with them.

D. Temperature Parameter ($\tau = 0.07$)

The temperature parameter controls the sharpness of the similarity distribution in contrastive learning. Similarity between two feature vectors is computed as

$$\text{similarity} = \frac{\exp(z_i \cdot z_j/\tau)}{\sum_k \exp(z_i \cdot z_k/\tau)}. \tag{9}$$

Lower values of τ produce a very sharp distribution in which only the most similar samples dominate the loss, while higher values flatten the distribution so that all samples contribute nearly equally. When τ is very small (e.g., 0.01), the similarity distribution becomes extremely peaked and sensitive to small differences. A medium value such as $\tau = 0.07$ yields moderate sharpness and is commonly used in contrastive learning. Larger values such as $\tau = 0.5$ result in a flat distribution where discrimination between samples is weak.

The choice of $\tau = 0.07$ is motivated by both empirical and practical considerations. This value is widely adopted in prior contrastive learning methods such as SimCLR, MoCo, and SupCon, and has been shown to work robustly across diverse domains. It also provides a good balance: the similarity function remains sharp enough to distinguish between similar and dissimilar samples, while avoiding overly extreme behavior that could destabilize training.

This behavior can be understood mathematically. For two very similar features with inner product $z_i \cdot z_j = 0.9$, the exponential term evaluates to

$$\exp(0.9/0.07) = \exp(12.86) \approx 383{,}000. \tag{10}$$

For a dissimilar pair with $z_i \cdot z_k = 0.1$, the exponential term becomes

$$\exp(0.1/0.07) = \exp(1.43) \approx 4.2. \tag{11}$$

The ratio between these values is therefore

$$\frac{383{,}000}{4.2} \approx 91{,}000\times, \tag{12}$$

indicating very strong discrimination between similar and dissimilar samples.

E. MoCo Design Rationale

A key design choice in MoCo is the use of a momentum update for the key encoder instead of standard gradient-based updates. Without momentum, features extracted at different iterations are generated by rapidly changing encoders. For example, at iteration t, features are encoded and added to the queue. At iteration $t{+}1$, the encoder parameters are updated and new features are generated, which are again added to the queue. As a result, the queue ends up containing features produced by different encoder states. This mismatch leads to incompatible representations and causes the contrastive loss to break down.

Momentum updates resolve this issue by ensuring that the key encoder evolves slowly over time. The update rule is given by

$$\theta_k = 0.999 \cdot \theta_k + 0.001 \cdot \theta_q. \tag{13}$$

With this formulation, the key encoder changes gradually. After 100 iterations, it has only evolved by approximately 10%, and after 1000 iterations it becomes about 63% aligned with the current query encoder. As a result, features stored in the queue remain compatible over time, preserving the integrity of the contrastive objective.

This consistency can be formally shown by unrolling the momentum update:

$$\theta_k^{(t)} = m^t \theta_k^{(0)} + (1 - m) \sum_{i=0}^{t-1} m^i \theta_q^{(t-i)}. \tag{14}$$

For $m = 0.999$, the key encoder is increasingly influenced by recent query encoder states while still maintaining smooth evolution. After 1000 steps, approximately 63% of the key encoder is influenced by recent parameters, and after 2000 steps this influence increases to about 86%, preventing abrupt changes and feature incompatibility.

Another important design decision is the use of a queue instead of simply increasing the batch size. Using a large batch incurs quadratic memory cost, since the similarity matrix scales as $O(B^2)$. In contrast, using a queue of size K leads to a linear memory cost of $O(B \cdot K)$ for query–queue similarity computation. For example, a batch size of 128 requires $128^2 = 16{,}384$ similarity pairs and approximately 1.2 GB of memory. Using a batch size of 32 with a queue of 2048 produces $32 \cdot 2048 = 65{,}536$ pairs while consuming only about 0.5 GB of memory. Thus, the queue provides roughly four times more negative pairs with approximately 60% less memory usage.

F. Mobile Home Exclusion: Statistical Justification

Excluding the mobile home class is methodologically justified due to its lack of test samples. In the dataset, this class has two training samples, zero validation samples, and zero test samples. As a result, precision and recall are both undefined during evaluation, since they involve divisions of the form $0/0$. Consequently, the F1 score is undefined and is conventionally assigned a value of 0.0000, even though the model is never actually evaluated on this class.

This has a direct and disproportionate impact on the macro F1 score. When mobile home is included as one of 17 classes, the macro F1 is computed as the average over all class-level F1 scores, including the artificial zero assigned to mobile home. In this case, the macro F1 becomes approximately $(16/17) \cdot 0.85 = 0.800$. When mobile home is excluded and the macro F1 is computed over 16 valid classes, the score becomes 0.85. The difference of 0.05 corresponds to a five–percentage–point drop that is entirely driven by a class with no test support.

This issue arises because the definition of macro F1 implicitly assumes that every class has nonzero support in the test set. When support is zero, precision, recall, and F1 are undefined, yet they still contribute to the macro average if included. This violates the assumption that all classes provide meaningful evaluation signals. Statistical best practice therefore recommends excluding classes with zero test support from macro averaging, or equivalently redefining macro F1 as

$$\text{Macro F1} = \frac{1}{C_{\text{valid}}} \sum_{c \in \text{valid}} \text{F1}_c, \tag{15}$$

where C_{valid} denotes the number of classes with nonzero test support.

Excluding mobile home also enables fair model comparison. When evaluated on 17 classes including mobile home, two models with macro F1 scores of 0.795 and 0.790 are difficult to distinguish, since both are equally penalized by the zero-score class. When evaluated on 16 valid classes, the same models achieve macro F1 scores of 0.847 and 0.810, respectively, revealing a clear performance difference of 3.7 percentage points. Therefore, excluding the mobile home class enables fair and meaningful comparison between models without distorting evaluation metrics. .

VII. Analysis and Insights

A. What Contrastive Learning Improves

Contrastive learning produces better feature clustering, where images with similar labels develop more similar feature representations. The feature space becomes more semantically structured, with related classes naturally grouping together. While t-SNE visualizations (not included in this report) demonstrate clearer class clusters compared to baseline models, the benefits extend beyond visualization to improved classification performance.

Rare class performance shows notable improvement through contrastive learning. Chaparral, while still challenging due to extreme data scarcity, benefits from feature similarity with vegetation-related classes. Court demonstrates measurable improvement in F1 score through better feature discrimination. Ship clusters with other water-related classes such as dock and sea, allowing it to leverage semantic relationships for improved predictions despite limited training examples.

The approach also provides robustness to label noise through its soft Jaccard weighting mechanism. Partial label overlap receives proportional weight rather than binary treatment, making the model less sensitive to individual label errors than hard classification approaches. This tolerance for imperfect labels can be valuable in real-world scenarios where annotation may be noisy or subjective.

B. What Contrastive Learning Doesn't Fix

Despite its benefits, contrastive learning cannot overcome extreme class imbalance. Chaparral with only 14 test samples continues to perform poorly with an F1 of 0.3784. While contrastive learning helps structure the feature space beneficially, it cannot overcome severe data scarcity. Addressing such extreme imbalance requires more training data or advanced augmentation techniques specifically targeting rare classes.

Confusion between visually similar classes persists under contrastive learning. Field versus grass confusion remains problematic, as these classes share similar visual characteristics. Similarly, chaparral exhibits continued confusion with other vegetation types. The contrastive loss, by design, groups visually similar objects together in feature space, which can actually reinforce confusion when different labels correspond to similar visual appearance. This represents a fundamental tension between visual similarity and semantic labels.

Small object detection remains challenging for airplanes and tanks that appear at small scales in aerial imagery. This limitation stems from the backbone architecture rather than the contrastive learning mechanism itself. The global average pooling in EfficientNet-B4 may lose spatial information necessary for detecting small objects. Addressing this would require architectural modifications such as multi-scale feature fusion or specialized detection heads, which fall outside the scope of contrastive representation learning.

C. Micro vs. Macro F1 Trade-off

Our contrastive learning approach achieves a macro F1 of 0.8425 and micro F1 of 0.9110 on the test set. Understanding the relationship between these metrics provides insight into model behavior across classes with different frequencies.

Micro F1, computed as:

$$\text{Micro F1} = \frac{2 \cdot \sum_{i,c} TP_{i,c}}{\sum_{i,c}(2 \cdot TP_{i,c} + FP_{i,c} + FN_{i,c})} \tag{16}$$

aggregates predictions across all samples and classes before computing the F1 score. This metric is dominated by frequent classes such as trees, pavement, and buildings, which contribute the majority of predictions. The micro F1 of 0.9110 indicates excellent overall prediction accuracy, weighted by class frequency.

Macro F1, computed as:

$$\text{Macro F1} = \frac{1}{C}\sum_{c=1}^{C} \text{F1}_c \tag{17}$$

treats all classes equally by averaging their individual F1 scores. The macro F1 of 0.8425 reflects balanced performance across all 16 classes, including rare classes that would be largely ignored in the micro F1 calculation.

Contrastive learning influences how model capacity is allocated across classes. Standard transfer learning optimizes primarily for overall accuracy, where frequent classes dominate gradient updates and drive learning. Contrastive learning creates a more balanced feature space where all classes receive structured representations regardless of frequency. This redistribution of model capacity tends to improve rare class performance (reflected in higher macro F1) while potentially causing minor decreases in frequent class performance (reflected in micro F1).

For imbalanced multi-label datasets, this trade-off is generally desirable. Macro F1 serves as the primary evaluation metric because it prevents frequent classes from dominating the assessment. Moreover, rare classes often carry greater practical value in real applications, such as detecting infrequent but important objects in aerial imagery for urban planning or environmental monitoring.

D. Loss Ratio as a Diagnostic Tool

The loss ratio serves as a valuable diagnostic for monitoring training dynamics. We track two ratios: the **unscaled ratio** $\mathcal{L}_{\text{contrastive}}/\mathcal{L}_{\text{BCE}}$ and the **scaled ratio** $(\lambda \cdot \mathcal{L}_{\text{contrastive}})/\mathcal{L}_{\text{BCE}}$. While the unscaled ratio indicates the inherent magnitude difference between loss functions, the scaled ratio determines actual training dynamics and relative gradient contributions.

Interpreting the scaled ratio requires understanding its implications. Scaled ratios below 0.5 indicate contrastive loss contributes minimally, providing limited representation learning benefits. Scaled ratios near 1.0 represent balanced training where both objectives receive approximately equal weight. Scaled ratios between 2-5 enter a zone where contrastive learning dominates, potentially overshadowing classification. Scaled ratios exceeding 10 indicate heavy contrastive dominance where the classification objective becomes secondary.

Our training trajectory reveals a dynamic shift in objective balance. The unscaled ratio increases dramatically from 8.2 at epoch 1 to 354.6 at epoch 35, driven by rapid BCE convergence (0.332 → 0.007, a 98% reduction) while contrastive loss decreases modestly (2.72 → 2.55, only 6% reduction). The scaled ratio consequently evolves from 0.41 (epoch 1: 71% BCE, 29% contrastive) to 17.7 (epoch 35: 5% BCE, 95% contrastive). This pattern suggests a two-phase training dynamic: early epochs focus on learning classification boundaries through BCE-dominated optimization, while later epochs refine feature space structure through contrastive-dominated learning.

Critically, despite the extreme ratio imbalance in later epochs, the model maintains strong classification performance (macro F1 = 0.8425), suggesting that contrastive learning does not degrade already-learned decision boundaries. This behavior can be interpreted as follows. Once the classifier achieves low BCE loss, gradient magnitudes from classification become small, allowing contrastive gradients to dominate without disrupting learned class separations. The contrastive objective then focuses on refining the geometric structure of the feature space, pulling together samples with shared labels while maintaining the discriminative boundaries established during early training. This staged learning process may actually be beneficial, as it prevents contrastive constraints from interfering with initial class boundary formation.

To contextualize our choice of $\lambda = 0.05$: With $\lambda = 0.5$, the scaled ratio would reach 177 at epoch 35, creating overwhelming contrastive dominance that might prevent effective classification learning. With $\lambda = 0.01$, the scaled ratio would be 3.5 at epoch 35, providing more balanced contributions but potentially underutilizing contrastive learning's representational benefits. Our selection of $\lambda = 0.05$ allows BCE-focused early learning (ratio 0.41) while enabling strong contrastive refinement in later stages (ratio 17.7), achieving a dynamic balance suited to the two-phase optimization process.

VIII. Summary

A. Work Completed

This work accomplished several technical achievements in applying contrastive learning to multi-label aerial imagery classification. We implemented supervised contrastive learning with Jaccard similarity weighting specifically adapted for the multi-label setting, achieving a macro F1 score of 0.8425 on 16 evaluable classes. We developed a MoCo variant that enables large-scale contrastive learning with 60% memory reduction compared to standard batch-based approaches, making the technique practical for resource-constrained environments. We identified and rigorously analyzed a critical dataset quality issue regarding the mobile home class, providing statistical justification for its exclusion from evaluation. Finally, we established proper loss balancing strategies through systematic experimentation with $\lambda = 0.05$ and continuous ratio monitoring.

Our research provides several contributions to the field of multi-label classification for remote sensing. We demonstrated contrastive learning's effectiveness for multi-label aerial classification, showing that implicit correlation modeling through feature space structuring can improve performance without explicit graph-based label modeling. We proved that Jaccard-based soft weighting outperforms binary overlap approaches for multi-label scenarios by capturing degrees of label similarity. We quantified the mobile home class impact on macro F1 evaluation, showing it artificially reduces scores by approximately 5.8 percentage points due to zero test samples. We validated MoCo as a practical solution for memory constrained contrastive training, demonstrating that queue-based negative sampling can provide 16x more negatives with 60% less memory. Finally, we provided comprehensive analysis of micro versus macro F1 trade-offs in imbalanced multi-label settings.

Our work delivers several practical outputs. The contrastive learning model achieves macro F1 of 0.8425 using EfficientNet-B4 with supervised contrastive learning. The MoCo model provides a memory-efficient alternative with 2048 negatives suitable for limited GPU resources. We provide complete, reproducible code for all experiments, along with detailed experimental analysis documenting training dynamics, per-class performance, and systematic ablation studies.

B. Limitations and Challenges

Our work faces several limitations that constrain performance and generalizability. Extreme class imbalance remains problematic, with a 24:1 ratio between the most and least frequent classes. Chaparral with only 14 test samples continues to challenge the model despite contrastive learning benefits. While weighted sampling helps address imbalance, it cannot fully overcome severe data scarcity, particularly for classes with fewer than 20 samples.

Computational requirements present practical constraints. Contrastive learning adds approximately 30% to training time compared to baseline transfer learning due to the additional projection head and similarity computations. MoCo reduces memory requirements but increases iterations per epoch, and queue maintenance introduces overhead that slows training. These factors make contrastive approaches more expensive than standard classification.

Hyperparameter sensitivity requires careful experimentation. The loss weight λ demands tuning to achieve proper balance between classification and contrastive objectives, with performance degrading significantly for poorly chosen values. Temperature τ impacts feature discrimination, with different values potentially optimal for different datasets or class distributions. MoCo introduces additional trade-offs between queue size and batch size that require dataset-specific optimization.

Dataset limitations fundamentally constrain achievable performance. With only 2,100 training images, the model has limited examples for learning robust representations, particularly for rare classes. The mobile home class is completely unusable due to having only 2 samples in the entire dataset. Furthermore, the dataset lacks multi-scale object annotations that could help address challenges with small objects like airplanes and tanks at distant scales.

C. Future Directions

Several promising avenues exist for further improving the contrastive learning approach. First, systematic hyperparameter tuning could yield significant gains, particularly by experimenting with different contrastive loss weights (0.1, 0.3, 0.7, 1.0) to find the optimal balance between classification and representation learning objectives. Similarly, temperature tuning in the contrastive loss function (exploring values such as 0.05, 0.1, and 0.2) could enhance the model's ability to distinguish between positive and negative pairs by controlling the concentration of the similarity distribution.

Architectural enhancements present another opportunity for advancement. Implementing hard negative mining would allow the model to focus computational resources on challenging negative pairs that provide stronger gradient signals, potentially leading to more discriminative feature representations. Additionally, adopting a MoCo-style momentum encoder could provide more stable and consistent features during training by maintaining a slowly-progressing key encoder. The current approach's limitation to within-batch positive pairs could be addressed by implementing a multi-positive contrastive learning strategy that leverages all available positive pairs across the dataset, though this would require careful memory management. Furthermore, combining the contrastive learning framework with Graph Convolutional Networks (GCNs) could explicitly model label co-occurrence patterns and semantic relationships, potentially capturing complex interdependencies between land cover classes. Finally, self-supervised pretraining on large-scale unlabeled aerial imagery datasets could provide superior initialization for the backbone network, particularly beneficial given the domain-specific nature of aerial image interpretation.

Comprehensive ablation studies would provide valuable insights into the contribution of individual design choices. Specifically, comparing binary overlap versus Jaccard similarity for defining positive pairs would quantify the benefit of soft label matching. Investigating different projection head dimensions (128, 256, 512) would help determine the optimal feature space dimensionality for contrastive learning in this multi-label context. Evaluating alternative backbone architectures such as Vision Transformers (ViT) or ConvNeXt could reveal whether attention-based or modern convolutional architectures better capture the hierarchical patterns in aerial imagery. Finally, systematic comparison of different contrastive loss variants including SimCLR, supervised contrastive loss (SupCon), and other recent formulations would identify the most effective approach for multi-label aerial image classification.

IX. Conclusion

This project successfully advanced multi-label aerial image classification through the application of supervised contrastive learning with Jaccard similarity-based weighting to explicitly structure the feature space according to label relationships. We achieved a macro F1 score of 0.8425 on 16 evaluable classes, demonstrating the effectiveness of contrastive learning for multi-label scenarios in aerial imagery.

Our work makes several key contributions. We identified and rigorously addressed a critical dataset quality issue where the mobile home class contained only 2 samples with zero test representation, artificially reducing macro F1 scores by approximately 5.8 percentage points. By excluding this statistically invalid class, we established a fair evaluation protocol for imbalanced multi-label datasets. We developed a memory-efficient MoCo variant enabling 16x more negative samples with 60% memory reduction, making large-scale contrastive learning practical even on resource-constrained hardware. Our theoretical analysis demonstrated that Jaccard-weighted soft contrastive loss outperforms binary overlap approaches for multi-label learning, and we established effective loss balancing strategies using $\lambda = 0.05$ with target ratio 1-5. Finally, we provided comprehensive analysis of micro versus macro F1 trade-offs, loss ratio diagnostics, and per-class performance patterns that offer insights into model behavior across imbalanced class distributions.

Contrastive learning particularly benefited challenging classes with limited training data. We observed improved feature clustering for low-support classes, better discrimination through label similarity-based feature structuring, and measurable F1 improvements on several minority classes. The approach successfully leveraged semantic relationships between labels to improve rare class predictions.

Our technical innovations include adapting Jaccard similarity weighting for multi-label contrastive loss, developing MoCo with label-aware queue management, employing loss ratio monitoring as a diagnostic tool for training stability, and conducting systematic dataset quality analysis that revealed important evaluation pitfalls.

The practical implications of this work extend beyond academic performance metrics. We demonstrate that modern contrastive learning techniques, originally designed for single-label classification, can be successfully adapted to multi-label aerial imagery with appropriate modifications. The combination of transfer learning, contrastive representation learning, and careful imbalance handling provides a strong foundation for real-world aerial scene understanding applications in urban planning, environmental monitoring, and disaster response.

The detailed per-class analysis, identified limitations, and proposed future directions offer valuable insights for both immediate improvements and long-term research in multi-label remote sensing classification. Future work on graph-based label correlation modeling, vision transformers, and ensemble methods provides a clear path toward further performance improvements while maintaining computational efficiency.

References

[1] Xia, G.-S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., Zhang, L., & Lu, X. (2017). AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7), 3965-3981.

[2] Hua, Y., Mou, L., & Zhu, X. X. (2020). Relation Network for Multilabel Aerial Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(7), 4558-4572.

[3] Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., & Xu, W. (2016). CNN-RNN: A Unified Framework for Multi-label Image Classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2285-2294.

[4] Chen, Z.-M., Wei, X.-S., Wang, P., & Guo, Y. (2019). Multi-Label Image Recognition with Graph Convolutional Networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5177-5186.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.

[6] Shabbir, A., Ali, N., Ahmed, J., Zafar, B., Rasheed, A., Sajid, M., Ahmed, A., & Dar, S. H. (2021). Satellite and Scene Image Classification Based on Transfer Learning and Fine Tuning of ResNet50. *Mathematical Problems in Engineering*, 2021, 5843816.

[7] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the International Conference on Machine Learning (ICML)*, 6105-6114.

[8] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised Contrastive Learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 18661-18673.

[9] He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum Contrast for Unsupervised Visual Representation Learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9729-9738.

[10] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. *Proceedings of the International Conference on Machine Learning (ICML)*, 1597-1607.

Appendix

Appendix: Contributions

All team members contributed equally to this project. The main responsibilities were:

Tunahan Yazar

- Contrastive learning implementation
- MoCo architecture
- Methodology section

Cansu Temizkan

- Dataset analysis
- Data preprocessing
- Dataset description section

Defne Koçulu

- Baseline model implementation
- Training experiments
- Results section

Yavuz Can Atalay

- Evaluation metrics
- Literature review, Research
- Introduction and conclusion

All members participated in debugging, report writing, and presentation preparation.