Meta Learning-based Few-Shot Learning: Remote Sensing

Priyanshu Gupta[1], *Shashwat Agarwal*[2], *Shantanu Gupta*[3]

The LNM Institute of Information Technology, Jaipur
Department of Computer Science and Engineering

[1]22UCS155, [2]22UEC123, [3]22UCC094

[1]22ucs155@lnmiit.ac.in, [2]22uec123@lnmiit.ac.in, [3]22ucc094@lnmiit.ac.in

April 8, 2024

Contents

1 Dataset in Meta Learning

1.1 EuroSAT Dataset

The **EuroSAT dataset** is a benchmark for land use and land cover classification using satellite imagery from the **Sentinel-2 satellite**, operated by the **European Space Agency (ESA)**. It was developed by **Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth**.

It contains **27,000 labeled images** of size **64 × 64 pixels** in both **RGB** and **multispectral** formats. The dataset includes **10 classes** such as *Annual Crop, Forest, Highway, Residential*, and *Sea/Lake*, serving as classification labels.

EuroSAT's strengths lie in its variety and quality, though it faces challenges like **class imbalance** and **limited resolution**. It remains widely used in *remote sensing* and *geospatial AI research.*

1.2 Aerial Image Dataset

The **AID (Aerial Image Dataset)**, developed by **Wuhan University**, is a large-scale benchmark for aerial scene classification. It includes **10,000 manually labeled RGB images** of size **600 × 600 pixels**, covering **30 diverse scene classes** such as *airport, forest, river, stadium*, and *residential areas*, with **220–420 samples per class**.

Collected from **Google Earth**, the images come from **multiple countries** (e.g., China, USA, UK), across various seasons, times, and lighting conditions, captured using **different remote sensors**, providing rich **intra-class diversity** and **high spatial resolution (0.5–8m)**. AID is widely used for benchmarking deep learning and aerial classification methods.

1.3 NWPU-RESISC45 dataset

The **NWPU-RESISC45 dataset**, developed by **Northwestern Polytechnical University**, contains **31,500 RGB aerial images** evenly distributed across **45 categories** (700 per class), including scenes like *airports, forests, stadiums*, and *residential areas.*

Each image is **256 × 256 pixels** in RGB format, sourced from **Google Earth**. The dataset features high **intra-class variation** in terms of resolution, lighting, viewpoint, and clarity. Images exhibit distinct **color, texture, spatial layout**, and **structural patterns**, with **196,608 features per image**.

NWPU-RESISC45 serves as a robust benchmark for **image classification, transfer learning**, and **domain adaptation** in remote sensing tasks. *(Leave this as a placeholder if you want to add more datasets.)*

2 Introduction to Meta Learning

Meta-learning, or "learning to learn," is an advanced paradigm in machine learning that enables models to rapidly adapt to new tasks using knowledge gained from a distribution of related tasks. It is especially useful in scenarios with limited labeled data, such as few-shot learning, where traditional approaches tend to fail.

2.1 Traditional Learning and Problems with It

Conventional machine learning (ML) and deep learning (DL) models rely heavily on large-scale labeled datasets. These models are trained from scratch for each task through two main phases:

- **Training Phase:** Models learn by optimizing parameters over extensive data.
- **Testing Phase:** Performance is evaluated on new, unseen data.

However, such models exhibit significant drawbacks:

- They lack the ability to generalize across different tasks.
- **Data distribution shift problem:** A mismatch between the training and testing distributions can cause a significant drop in performance.
- They are ineffective in scenarios where large datasets are unavailable.

2.2 Transfer Learning vs. Meta-Learning

Transfer learning aims to reuse knowledge from a source task to improve learning on a target task, typically by:

- **Model Parameters:** Adjusting pretrained weights for a new task.
- **Feature Representations:** Using patterns learned from one task to recognize similar patterns in another.
- **Instances (Data Points):** Applying past data to inform new learning problems.

While transfer learning improves performance for related tasks, it struggles when:

- The target dataset is extremely limited.
- There is a significant domain shift between source and target tasks.

In contrast, meta-learning learns the learning process itself, allowing models to quickly adapt to new tasks using prior task experience rather than relying solely on data similarity.

2.3 What is Meta-Learning?

Meta-learning frameworks are trained over a variety of tasks, not just one. Rather than optimizing only model parameters, they aim to optimize for rapid adaptability. In practice, meta-learning operates over **episodes**, where:

- Each episode consists of a **support set** (training data) and a **query set** (test data).
- The model adapts to the task using the support set and is evaluated on the query set.

The central idea is to extract transferable knowledge across tasks, enabling the model to perform well on unseen tasks with minimal data.

Formally, according to Thrun and Pratt:

> "A model improves its ability to solve a set of tasks by gaining experience from multiple related tasks."

This stands in contrast to traditional learning, where models are typically trained to solve a single task at a time.

2.4 Why Meta Learning?

Meta-learning offers several critical advantages:

- **Rapid Adaptation:** It enables models to learn from a few examples with minimal parameter updates.
- **Improved Generalization:** Models trained on diverse tasks can learn robust patterns that transfer effectively to new, unseen domains.
- **Data Efficiency:** Meta-learning significantly reduces reliance on large annotated datasets, making it ideal for low-data regimes.
- **Human-Like Learning:** It mimics the way humans leverage past experiences to accelerate learning in new situations.

Meta-learning serves as the foundation for few-shot learning, where training is conducted in an **episodic manner** that mirrors real-world testing conditions. This alignment makes meta-learning particularly applicable in practical AI systems where data is scarce, expensive to label, or rapidly changing.

3 Components of Meta Learning

3.1 Metric-Based Methods

Metric-based meta-learning models focus on learning a similarity function that can be used to compare query samples with support examples. These models are particularly useful in few-shot learning scenarios.

3.1.1 Siamese Networks

Siamese networks are one of the earliest models designed for few-shot learning. They are composed of twin networks that share weights and are trained to differentiate between pairs of inputs.

How it works:

- A pair of images is passed through two identical CNNs to obtain feature embeddings.
- A distance metric (e.g., Euclidean distance or cosine similarity) is used to measure the similarity between the embeddings.
- If the distance is small, the images are considered similar; otherwise, they are considered different.

Loss Function:

- Contrastive loss or binary cross-entropy is used to train the model.

Training:

- During training, the model sees both positive (same class) and negative (different class) image pairs.

Few-Shot Inference:

- During testing, a query image is compared to all support images.
- The label of the most similar support image is assigned to the query.

3.1.2 Prototypical Networks

Prototypical Networks represent each class by the mean of its embedded support examples, known as a prototype.

Key Steps:

- Embed each image using a neural network.
- Compute the mean embedding (prototype) for each class.
- Classify query points based on distances to class prototypes using Euclidean or cosine similarity.

Loss Function:

- Cross-entropy loss with softmax over negative distances.

Advantage:

- Simplicity and efficiency. Works well for few-shot classification.

3.1.3 Matching Networks

Matching networks help classify new examples with few labeled examples.

Mechanism:

- **Support Set & Query Point:** Trained with a small support set $S = \{(x_i, y_i)\}$.
- **Attention Mechanism:** Computes cosine similarity between query and support set embeddings.
- **Prediction:** Assigns label using weighted sum of support labels based on similarity.

Variations:

- **CFMN:** Matches image regions using spatial attention.
- **CCL:** Improves inter-class separation.
- **Hard Label Training:** Learns from increasingly difficult examples.

3.1.4 Relation Networks

Relation Networks are composed of:

- **Embedding Function (f_ϕ):** Extracts features from images.
- **Relation Module (g_ϕ):** Learns to compute similarity between embeddings.

Workflow:

- Convert support and query images to embeddings.
- Average embeddings for each class.
- Compare query to class embeddings using relation module.
- Use MSE loss to learn relation scores.

Extensions:

- **MRN:** Adds memory from similar images.
- **Prototype-Relation Network:** Combines prototype averaging and relation learning.
- **MSRN:** Uses features from multiple CNN layers.
- **MLFRNet:** Focuses on local features via cropping.
- **MsKPRN:** Handles object size/position variation using multi-scale encoding.

3.1.5 Advanced Metric-Based Meta-Learning Methods

1. Graph Neural Networks (GNN):

- Unifies Siamese, Prototypical, and Matching networks.
- Treats tasks as graphs with nodes (support embeddings + labels) and edges (relations).
- Trained with cross-entropy loss.

2. Global Class Representation (GCR):

- Builds a global class space from base and novel classes.
- Uses registration loss and episodic prototypes.

3. Multi-Scale Features and Non-Metric Similarity:

- Extracts multi-scale features.
- Uses learned non-metric similarity functions.

4. Attentive Metric-Based Learning:

- **ARCs:** Sequential region focus with RNN and attention.
- **RCN:** Compares specific image regions dynamically.

5. MACO (Metric-Agnostic Conditional Embeddings):

- Avoids fixed similarity metrics.
- Learns similarity via fully connected layers.

3.2 Memory-Based Methods

Memory-based meta-learning methods utilize memory components to store and retrieve information from past tasks, enabling fast adaptation to new learning tasks. There are two types of memory used:

- **External Memory:** Analogous to writing down notes in a notebook.
- **Internal Memory:** Like retaining information in your brain.

3.2.1 Memory-Augmented Neural Networks (MANN)

- **Inspiration:** Similar to taking notes while studying and referring back to them.
- **Neural Turing Machine (NTM):** Core structure with an external memory bank.
- **Key Components:**
  - **Controller:** Determines what and how to read/write.
  - **Memory Bank:** A 2D matrix to store task-related information.

  - **Read/Write Mechanisms:** Includes erasing and updating memory slots.
  - **Addressing Mechanisms:**
    * **Content-based:** Access based on similarity.
    * **Location-based:** Sequential access to memory slots.

- **Example:** Learning foreign language words by writing and revisiting similar examples.

3.2.2 Least Recently Used Access (LRUA)

- **Motivation:** Prevents overwriting recently used information.
- **Strategy:** Writes to memory locations that have not been accessed for the longest time.
- **Analogy:** Similar to a phone clearing apps that haven't been used recently.

3.2.3 Simple Neural Attentive Meta-Learner (SNAIL)

- **Memory Type:** Internal memory system (no external memory bank).
- **Core Components:**
  - **Temporal Convolution (T-CNN):** Enables fast access to past knowledge.
  - **Soft Attention Mechanism:** Highlights important parts of past experiences.
- **Analogy:** Like remembering traffic rules instantly when seeing a red light.

3.2.4 Conditional Neural Processes (CNPs)

- **Concept:** Create compact summaries of prior tasks instead of storing raw memory.
- **Key Modules:**
  - **Embedding Function (h):** Maps input data to feature representations.
  - **Aggregation Function (a):** Combines embeddings into a summary.
  - **Task Learner (g):** Uses the summary to make predictions on new tasks.
- **Analogy:** Classifying dog breeds using key summarized traits instead of memorizing all examples.

3.2.5 Memory-Augmented Matching Networks

- **Problem Addressed:** Bias from uneven class sizes in prototypical averaging.
- **Solution:**
  - **Weighted Class Prototypes:** More representative samples get higher weights.
  - **Distance-Based Weights:** Closer samples contribute more to the prototype.
- **Analogy:** Grouping math learners based on true proficiency rather than average scores.

3.3 Learning-Based Methods

Learning-based meta-learning aims to train a meta-learner that directly influences the learning dynamics of a base learner. Instead of just optimizing model weights for a specific task, it learns *how to learn* by observing how models perform across multiple tasks. The key idea is to extract transferrable learning rules, parameter initializations, or optimization strategies that can be reused to rapidly adapt to new tasks with minimal data.

These approaches differ in what component they learn: the optimizer, the model initialization, or the parameters themselves.

3.3.1 Learning the Initialization

Learns a good initial point in the parameter space that can be quickly fine-tuned for new tasks using few gradient updates.

- **MAML**: Finds a shared initialization that generalizes well after a few steps of gradient descent.
- **LEO**: Learns low-dimensional latent embeddings for initialization to improve generalization.
- **PLATIPUS, BMAML**: Introduce uncertainty modeling through Bayesian formulations.
- **BOIL, TAML, CAML, ADML**: Enhance robustness using inner-loop feature updates, task-agnostic losses, context vectors, or adversarial training.

3.3.2 Learning the Parameters

Trains a meta-network to generate the weights or parameters of a task-specific learner directly.

- **MetaNet**: Generates task-specific weights using a fast weight mechanism.
- **LGM-Net**: Learns to produce classifier parameters via context encoding and weight generation.
- **Weight Imprinting**: Assigns weights for novel classes using normalized class embeddings.
- **DAE, TAFE-Net, MTL**: Use denoising, task-awareness, or transfer learning to improve parameter estimation.

3.3.3 Learning the Optimizer

Learns the optimization algorithm itself to enable faster and more adaptive learning.

- **LSTM Optimizer**: Uses an RNN to learn how to update weights over time, simulating gradient descent.
- **Meta-SGD**: Learns both the initialization and a task-specific learning rate.
- **RNNProp**: Mimics gradient-based learning rules using recurrent architectures.

4 Challenges in Meta Learning

4.1 Divergent Optimization Directions

Meta-learners often fail to generalize when training tasks differ significantly from test tasks, leading to ineffective adaptation.

Solutions:

- Train on diverse and realistic tasks.
- Group similar tasks for better learning.
- Use **Bayesian Active Learning (BAL)** to select informative samples.
- Apply **Variational Autoencoders (VAEs)** to capture task structure.

4.2 Catastrophic Forgetting

Models forget earlier tasks when adapting to new ones, harming base class performance.

Solution:

- Combine **memory-based** (retains past knowledge) and **metric-based** (task similarity) methods.

4.3 Dataset Instability

Models perform inconsistently across datasets (e.g., good on Omniglot, poor on MiniImageNet).

Solution:

- Improve generalization strategies for varied datasets.

4.4 Adversarial Vulnerability

Meta-learners like MAML are sensitive to adversarial inputs.

Solutions:

- Use **RCN** for better feature extraction.
- Explore **ADML** to enhance robustness.

4.5 Poor Cross-Domain Performance

Single-domain training limits generalization to new domains.

Solutions:

- Adopt **multi-domain training**.
- Use realistic datasets (e.g., remote sensing, agriculture).

4.6 Architectural Choices

Lack of clarity on how backbone networks affect performance.

Solutions:

- Compare CNNs, ResNet, and multi-scale networks.
- Use **MC Dropout** for uncertainty estimation.
- Combine with **Meta-SGD** for dynamic learning.

5 Future Research in Meta Learning

To address current limitations, future meta-learning research is focusing on the following areas:

5.1 Hybrid Models

- Combine meta-learning with deep learning techniques.
- Example: Merge memory-based (retains past knowledge) and metric-based (compares task similarity) methods to boost performance.

5.2 Multi-Domain Learning

- Train models across diverse domains (e.g., X-ray, MRI, CT).
- Enhances adaptability to different environments and devices.

5.3 Model Explainability

- Use Explainable AI (XAI) to interpret model decisions.
- Example: Highlight abnormal regions in medical images for transparency.

5.4 Reducing Computational Costs

- Develop efficient training strategies for resource-limited environments.
- Example: Apply Neural Architecture Search (NAS) to automate model optimization.

5.5 Enhancing Robustness

- Improve model stability and reliability under noisy or uncertain inputs.
- Crucial for real-world applications with imperfect data.

6 Applications of Meta Learning

6.1 Computer Vision

In computer vision, meta-learning is especially useful when image data is limited. It has several applications such as:

- **Remote Sensing Scene Classification:** Meta-learning helps in analyzing satellite images to identify land types like forests, water bodies, cities, or farmland. This is useful in environmental monitoring and disaster management.
- **Medical Image Classification:** It is used to detect diseases from images such as X-rays or MRI scans where labeled data is often scarce.
- **Object Detection and Image Segmentation:** Meta-learning enables the identification and separation of objects in images, like locating a tumor in a medical scan or detecting vehicles in road surveillance.

6.2 Language and Speech Processing

Meta-learning improves the way machines understand human language, especially when there are only a few training examples available. Key applications include:

- **Text Classification and Sentiment Analysis:** It helps sort text into topics (e.g., sports, politics) and find emotional tone in reviews (positive, negative, neutral).
- **Machine Translation:** It allows translating languages even with small amounts of data, making it helpful for rare or low-resource languages.
- **Speech Recognition:** Meta-learning helps convert spoken words into text and adjust to various accents or speaking styles.

6.3 Reinforcement Learning and Robotics

In reinforcement learning and robotics, meta-learning helps intelligent systems adapt to changing environments quickly. Major uses include:

- **Quick Adaptation to New Environments:** AI agents and robots can learn new tasks or adjust to new surroundings without retraining from scratch.
- **Handling Unexpected Changes:** Robots can respond to sudden problems, such as a motor failure, by learning from a few examples and adjusting their actions.

7 Conclusion

Meta learning, also known as "learning to learn," is a helpful method for solving problems where we don't have enough labeled data—especially in remote sensing. Traditional deep learning needs a lot of training data, which is hard to get in this field due to limited images, high labeling costs, and class imbalance.

Few-shot learning, supported by meta learning methods like metric-based, memory-based, and learning-based approaches, helps models quickly adapt to new tasks with very few examples. This is useful in remote sensing, where we often deal with new places or scenes that were not seen during training.

We studied popular datasets like **EuroSAT**, **AID**, and **NWPU-RESISC45** to understand how diverse and detailed data helps test the performance of few-shot learning models. As more remote sensing data becomes available, meta learning can help create models that are more flexible, powerful, and able to work in many different situations.

In the future, researchers can look into new areas like *unsupervised meta learning*, using time-based data, and building models that keep learning and improving as they get more information.