# Research Proposal: A Framework for Long-Tailed Image Classification

## 1. Introduction

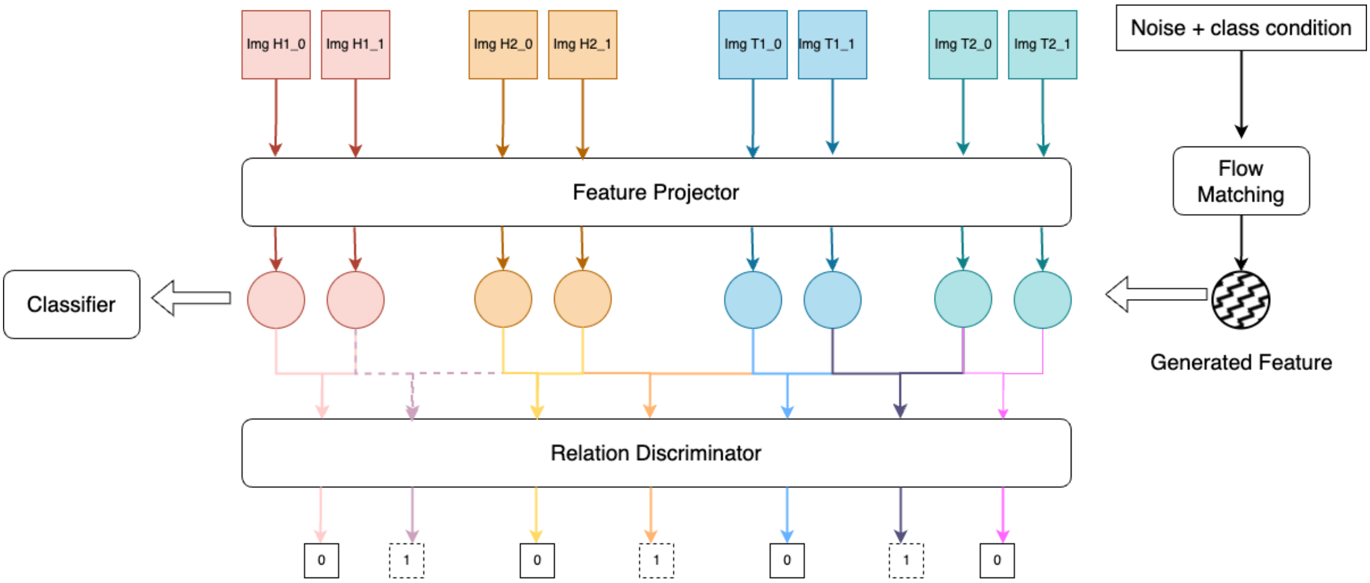
Long-tailed image classification remains a formidable challenge in computer vision. The skewed class distribution, with a handful of head classes having abundant samples and numerous tail classes suffering from data scarcity, severely hampers the performance of deep learning models. These models often overfit to head classes while underperforming on tail classes, leading to subpar generalization. Existing solutions typically rely on complex manual adjustments, like tuning angular variance, to homogenize feature space distributions across classes. Inspired by our RelationGAN paper, this research introduces a novel framework that automates and optimizes these processes for enhanced long-tailed image classification.

## 2. Research Objectives

The core aim of this research is to devise an advanced framework for long-tailed image classification, centered around two key objectives:

1. Leverage an adversarial training mechanism between a feature projector and a relation discriminator, informed by the RelationGAN concept and utilizing triplet loss, to align the feature distributions of tail and head classes within the same category, eliminating the need for manual angular variance tuning.
2. Employ a flow matching model to learn and generate the distribution of projected features from class conditions and noise, effectively augmenting the scarce data of tail classes and boosting their classification performance.

## 3. Proposed Framework



## 3.1 First Component: Feature Projector and Adversarial Relation Discriminator

- **Feature Projector**: The feature projector takes images as input and is typically constructed using a deep neural network architecture, such as a Convolutional Neural Network (CNN). It can be pre-trained on a large-scale general-purpose image dataset and then fine-tuned for the long-tailed image classification task. Given an input image

  The feature projector takes images as input and is typically constructed using a deep neural network architecture, such as a Convolutional Neural Network (CNN). It can be pre-trained on a large-scale general-purpose image dataset and then fine-tuned for the long-tailed image classification task. Given an input image $I$, the feature projector $P$ outputs a corresponding feature vector $f = P(I)$. Its role is to extract discriminative features that capture the essence of the input images, serving as the foundation for subsequent processing.

- **Adversarial Relation Discriminator**: Inspired by the RelationGAN approach detailed in the reference paper, the relation discriminator

  Inspired by the RelationGAN approach detailed in the reference paper, the relation discriminator $D$ takes triplets of feature vectors $(f_a, f_p, f_n)$ as input, where $f_a$ is the anchor feature, $f_p$ is a positive feature from the same class as the anchor, and $f_n$ is a negative feature from a different class. Its objective is to distinguish the similarity relationships among these features.

During training, we adopt triplet loss, which aims to maximize the distance between the anchor and negative features while minimizing the distance between the anchor and positive features. The triplet loss function $L_{triplet}$ is defined as:

$$L_{triplet} = \max\{d(f_a, f_p) - d(f_a, f_n) + \alpha, 0\}$$

where $d$ represents a distance metric, such as Euclidean distance, and $\alpha$ is a margin parameter that controls the minimum distance between the positive and negative pairs.

The feature projector and relation discriminator engage in adversarial training. The relation discriminator is trained to accurately identify valid triplets based on the triplet loss, while the feature projector aims to project features in a way that confounds the discriminator, making it difficult to distinguish correct similarity relationships. This adversarial process drives the feature projector to make the feature space distribution of each class more uniform, particularly reducing the gap between tail and head class features within the same class.

### 3.2 Second Component: Flow Matching Model;

- **Input**: The flow matching model takes class conditions;

  The flow matching model takes class conditions $c$ and noise $n$ as inputs. Class conditions can be encoded as one-hot vectors or other suitable representations to specify the target class for feature generation. The noise, often in the form of random Gaussian noise, adds variability to the generated features.

- **Learning the Distribution**: The flow matching model;

  The flow matching model $F$ endeavors to learn the distribution of the projected features from the feature projector. It is trained to generate feature vectors $\hat{f}$ such that $\hat{f} = F(c, n)$ closely approximates the distribution of the actual projected features $f$ for the corresponding class. This is achieved through techniques like normalizing flows, which transform simple probability distributions into more complex ones.

The training of the flow matching model is guided by minimizing the Kullback-Leibler (KL) divergence between the distribution of the generated features $p_{gen}(\hat{f}|c)$ and the true distribution of the projected features $p_{true}(f|c)$ for each class. The loss function $L_F$ for the flow matching model is:

$$L_F = \sum_c KL(p_{gen}(\hat{f}|c)||p_{true}(f|c))$$

Once trained, the flow matching model can generate a plethora of features for tail classes. These generated features are then used to augment the training data, enabling the overall long-tailed image classification model to better capture the characteristics of tail classes.

## 4. Methodology

### 4.1 Data Preparation;

Select well-established long-tailed image datasets, such as the iNaturalist 2018 dataset or the Places-LT dataset, known for their imbalanced class distributions. Split the datasets into training, validation, and test sets in a stratified manner to ensure each subset maintains a representative proportion of samples from all classes, facilitating fair model evaluation and training.

### 4.2 Training the Feature Projector and Relation Discriminator;

Initialize the feature projector and relation discriminator with appropriate weights, leveraging pre-trained weights for the feature projector when applicable. In each training iteration, sample triplets of images from the training set. For each triplet, obtain the corresponding feature vectors using the feature projector. Update the relation discriminator by minimizing the triplet loss, and then update the feature projector to maximize the triplet loss with respect to its parameters. Employ techniques like mini-batch training and gradient descent optimization algorithms, such as the Adam optimizer, to ensure efficient and stable training of these components.

### 4.3 Training the Flow Matching Model;

Initialize the flow matching model, basing its architecture on existing normalizing flow designs like Real-NVP or Glow. During training, sample class conditions and noise vectors, generate feature vectors using the flow matching model, and compute the KL divergence loss between the generated and true feature distributions for each class. Update the model's parameters using gradient-based optimization to minimize the KL divergence loss, enabling the model to learn and generate accurate feature distributions.

### 4.4 Combining the Components for Long-Tailed Image Classification;

After training the feature projector, relation discriminator, and flow matching model, use the feature projector to extract features from the training and test images. Augment the training data for tail classes by generating additional features with the flow matching model. Train a final classifier, such as a fully-connected neural network classifier, on the augmented training data. The classifier takes the projected features as input and outputs class predictions. Evaluate the performance of the entire framework on the test set using comprehensive metrics, including accuracy, precision, recall, and F1-score for each class, with a particular focus on the improvement in tail class performance.

## 5. Conclusion

This research proposal presents an innovative framework for long-tailed image classification that integrates an adversarial training mechanism inspired by RelationGAN with triplet loss and a flow matching model for feature generation. By directly addressing the challenges of feature distribution alignment and data scarcity in tail classes, this framework holds great promise for advancing the field. The detailed methodology provides a clear roadmap for implementation and evaluation, and we are eager to conduct extensive experiments to validate the framework's efficacy.