Model Architecture and Training Summary

Pretrained Model Selection

We selected two pretrained CNN architectures from the PyTorch model zoo:

- **ResNet18**
- **MobileNetV2**

These models were chosen due to their balance between representational power and computational efficiency for low-resolution inputs.

Input Preprocessing

Property	Setting
Image Size	64×64 pixels
Color Channels	3 (RGB)
Normalization	Mean: [0.485, 0.456, 0.406] Std: [0.229, 0.224, 0.225]
Augmentation	None (for consistency in few-shot setup)

Few-Shot Learning Setup

Parameter	Value
N-way	5
K-shot	1 and 5
Query images	5 per class
Sampling method	Random class and image selection per episode
Test episodes	100 (standardized for fair comparison)

Model Architecture

Both models use their convolutional layers as feature encoders:

ResNet18

- Used all layers up to the global average pooling layer

- Final classifier layer removed

MobileNetV2 (Used in Final Model)

- Used mobilenet.features
- Added AdaptiveAvgPool2d((1,1)) to generate compact prototype embeddings

---

Training Configuration

Setting	Value
Optimizer	Adam
Learning Rate	1e-4
Loss Function	CrossEntropy (on negative Euclidean distances)
Training Episodes	100 per epoch
Epochs	10
Framework	PyTorch
Training Device	CPU

Validation Strategy

- **Few-shot episodic evaluation** was used on the Mini-ImageNet test set.
- Results were averaged over **100 random episodes** to ensure robustness.

Final Performance (on Test Set)

Setup	Mean Accuracy	Std Deviation
5-way 1-shot	50.9%	±11.2%
5-way 5-shot	67.4%	±11.9%