> The goal of this work is to devise a few-shot visual learning system that during test time it will be able to efficiently learn novel categories from only a few training data while at the same time it will not forget the initial categories on which it was trained (here called base categories).

## Summary

**Propose**:

- To extend an object recognition system with an attention based few-shot classification weight generator.
- To redesign the classifier of a ConvNet model as the cosine similarity function // between feature representations and classification weight vectors. (apart from unifying the recognition of both novel and base categories, it also leads to feature representations that generalize better on “unseen” categories. )

**实验数据集**:

- Mini-ImageNet：

**实验结果**:

- 1-shot: 56.20%
- 5-shot: 73.00%

**Other**:

> - jupyter notebook： Github-Code
> - 数据集： Link
> - 论文： Dynamic Few-Shot Visual Learning without Forgetting
> - 源码： Github-Code
> - 相关博客：
>   - few-shot learning是什么

---

## Few-shot learning 的要求

- the learning of the novel categories needs to be fast （快速学习新事物）
- to not sacrifice any recognition accuracy on the initial categories that the ConvNet was trained on, i.e., to not “forget” （不忘记旧事物）

## 本文提出两个新的技术

### 1、Few-shot classification-weight generator based on attention.

**传统的图像分类方法**： 先提取图像的高维特征，然后通过分类器计算属于每一个类别的概率（这个概率向量这里成为 “分类权重向量（classification weight vectors） ”）.

这里使用一个额外的组件 “few-shot classification weight generator”， 在接受新的事物时（1-5个新类别），生成新的分类权重向量。主要特征是：通过将注意力机制纳入基本类别的分类权重向量上，从而显式地利用了过去获得的有关视觉世界的知识。

## 2、Cosine-similarity based ConvNet recognition model.

xxx

# Methodology

定义一个 具有 $K_{base}$ 个基本类别的数据集：

$$D_{train} = \bigcup_{b=1}^{K_{base}} \{x_{b,i}\}_{i=1}^{N_b} \tag{1}$$

where $N_b$ is the number of training examples of the $b$ -th category and $x_{b,i}$ is its $i$ -th training example.

本文的目标是： 使用此数据集作为唯一输入，既能够准确地识别基本类别，又能在不忘记基本类别的前提下，动态地经过少量样本学习来识别新的类别。

## Model 总览

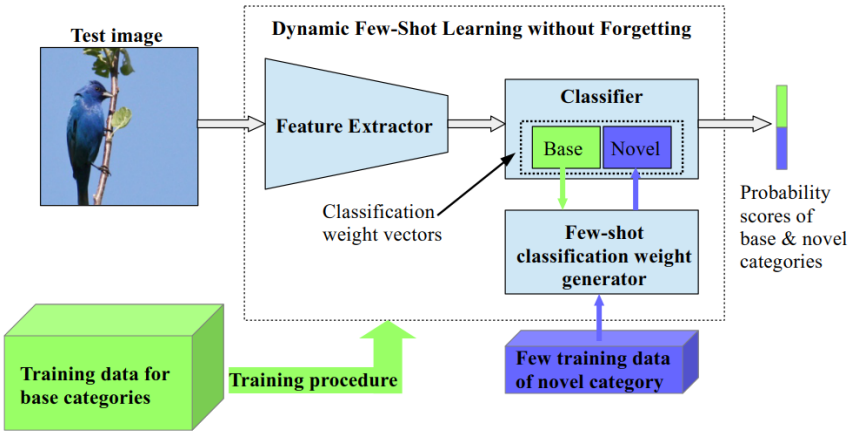


**Figure 1:** Overview of our system. It consists of: (a) a *ConvNet based recognition model* (that includes a feature extractor and a classifier) and (b) a *few-shot classification weight generator*. Both are trained on a set of base categories for which we have available a large set of training data. During test time, the weight generator gets as input a few training data of a novel category and the classification weight vectors of base categories (green rectangle inside the classifier box) and generates a classification weight vector for this novel category (blue rectangle inside the classifier box). This allows the ConvNet to recognize both base and novel categories.

### ConvNet-based recognition model

- A **feature extractor** $F(.|\theta)$ (with learnable parameters $\theta$) that extracts a $d$ -dimensional feature vector $z = F(x|\theta) \in \mathbb{R}^d$ from an input image $x$,
- A **classifier** $C\left(.|W^*\right), \$where\$W^* = \left\{w_k^* \in \mathbb{R}^d\right\}_{k=1}^{K}$ are a set of $K^*$ classification weight vectors - one per object category, that takes as input the feature representation $z$ and returns a $K^*$ -dimensional vector with the probability classification scores $p = C\left(z|W^*\right) \$of the\$K^*$ categories.

(也就是传统的分类模型的两个模块)

We learn the $\theta$ parameters and the classification weight vectors of the base categories $W_{base} = \{w_k\}_{k=1}^{K_{base}}$ such that by setting $W^* = W_{base}$ the ConvNet model will be able to recognize the base object categories.

### Few-shot classification weight generator

> Meta-learning mechanism. 在测试时，接受 $K_{novel}$ 个新类别的少量数据作为输入。

$$D_{novel} = \bigcup_{n=1}^{K_{novel}} \left\{ x'_{n,i} \right\}_{i=1}^{N'_n} \tag{3}$$

where $N'_n$ is the number of training examples of the $n$ -th novel category and $x'_{n,i}$ is its $i$ -th training example.

> - novel category $n \in [1, N_{\text{novel}}]$
> - few-shot classification weight generator $G(.,..|\phi)$
> - input the feature vectors $Z'_n = \left\{ z'_{n,i} \right\}_{i=1}^{N'_n}$
> - training examples $N'_n$
> - here $z'_{n,i} = F\left( x'_{n,i} | \theta \right)$
> - classification weight vector $w'_n = G\left( Z'_n, W_{base} | \phi \right)$

简单来说就是使用预训练好的特征提取器提取新类别图像的特征，然后将其扔进 few-shot classification weight generator 训练， 得到 classification weight vector， 得到 $W_{\text{novel}} = \{w'_n\}_{n=1}^{K_{novel}}$ (the classification weight vectors of the novel categories inferred by the few-shot weight generator). 最后，在 $C\left(. | W^*\right)$ 中合并两个分类权重向量： $W^* = W_{base} \cup W_{novel}$ ， 使 ConvNet 可以同时分类出 base 和 novel 中的类。

## Model 详细

### Cosine-similarity based recognition model

关于怎么在测试时合并可变数量的新的类别。