

### **Beginner level: Fine-grained image classification:**

The Oxford Flowers 102 dataset is a consistent of 102 flower categories commonly occurring in the United Kingdom. Each class consists of between 40 and 258 images. The images have large scale, pose and light variations. In addition, there are categories that have large variations within the category and several very similar categories.

<https://paperswithcode.com/dataset/oxford-102-flower>

The Stanford Dogs dataset contains images of 120 breeds of dogs from around the world. This dataset has been built using images and annotation from ImageNet for the task of fine-grained image categorization. There are 20,580 images.

<https://paperswithcode.com/dataset/stanford-dogs>

Caltech-101 consists of pictures of objects belonging to 101 classes, plus one background clutter class. Each image is labelled with a single object. Each class contains roughly 40 to 800 images, totalling around 9k images. Images are of variable sizes, with typical edge lengths of 200-300 pixels. <https://paperswithcode.com/dataset/caltech-101>

Code to start with:

[https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/vision/ipynb/image\\_classification\\_efficientnet\\_fine\\_tuning.ipynb](https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/vision/ipynb/image_classification_efficientnet_fine_tuning.ipynb)

### **Basic level:**

Start with the beginner level project. Build on top of it.

Combine the benefit of Vision Transformers and Convolutional Neural Networks for the fine-grained image classification.

Code for help:

<https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/vision/ipynb/mobilevit.ipynb>

### **Advanced level:**

CV Idea: Unsupervised or semi-supervised domain adaptation on Digit-Five datasets (Hull, MNIST, MNIST-M, SVHN, Synthetic number)

Code:

<https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/vision/ipynb/adamatch.ipynb>

NLP Idea: Finetune large language models (LLMs) like GPT in a parameter efficient manner for text generation

Code to start with:

[https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/nlp/ipynb/parameter\\_efficient\\_finetuning\\_of\\_gpt2\\_with\\_lora.ipynb](https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/nlp/ipynb/parameter_efficient_finetuning_of_gpt2_with_lora.ipynb)

### **Task for week 1:**

Choose your level, project, and go through the code, datasets provided.

Read and write a one page summary of the method based on the given code.

Read and write a one page summary of the dataset description and the problem formulation.