PRACTICAL TECH GUIDE

AI Image Classification

Train Models Locally — A Hands-On Guide

red_deer 33%, elk 21%, white_tailed 18%

AI Image Classification

Train Models Locally — A Hands-On Guide

Version 1.0

Disclaimer

This book is intended for educational and informational purposes only.

Tools and commands may change over time.

Table of Contents

PART ONE: GETTING STARTED

Chapter 1 — What This Book Is About

This book shows you how to train your own image classification model — a program that looks at a photo and tells you what it contains. You will do it on your own computer, with your own data, using free and open-source tools.

No cloud accounts. No monthly fees. No machine learning PhD required.

By the end, you will understand the full process from collecting images to deploying a working model — and you will have done it yourself.

Who this book is for

You do not need a background in machine learning. If you can install Python and run a command in a terminal, you have everything you need to follow this guide from start to finish.

It is also useful if you have dabbled in ML before and want a clear, practical reference for preparing datasets, choosing architectures, and understanding training metrics without wading through academic papers.

What you will be able to do by the end

- *Prepare a labelled image dataset from your own photos or an existing collection.*
- *Launch a training run and watch it progress in real time.*
- *Understand what accuracy, loss, and validation mean — and use them to make better decisions about your model.*
- *Choose the right model architecture for your needs.*
- *Export your trained model and integrate it into an application.*
- *Decide when local training makes sense and when a cloud service is a better fit.*

What this book does not cover

This is not a textbook on neural network theory. We explain how things work at the level that helps you make good practical decisions — without derivations, heavy notation, or mathematical proofs. We also do not cover object detection or image segmentation; those are separate (and more complex) problems.

About the companion tool

This book is built around an open-source project called the Image Classification Trainer — a browser-based web UI that wraps two training engines (PyTorch via timm, and TensorFlow via Keras) behind a simple interface. You pick your dataset, choose an architecture, set a few numbers, and click Start.

The source code is at:

https://github.com/nextframedev/image-classification-trainer

Where we show code snippets, they come from that project. Every concept in this book applies equally if you write your own training script or use a different tool — the companion project just makes it easier to follow along.

Chapter 2 — Quick Start: Your First Model in Minutes

This chapter gets you running immediately. Later chapters explain the why behind each step.

Requirements

- Python 3.10 or newer (python.org/downloads)
- A terminal (Terminal on macOS/Linux; PowerShell on Windows)
- At least 4 GB of RAM; 8 GB or more is comfortable
- A dataset of labelled images (see Chapter 6 if you do not have one yet; for now any folder of images sorted into subfolders by category will work)

Step 1: Get the code

git clone https://github.com/nextframedev/image-classification-trainer
cd image-classification-trainer

Step 2: Create a Python virtual environment

A virtual environment keeps this project's dependencies separate from everything else on your system.

python3 -m venv venv
source venv/bin/activate # macOS / Linux
venv\Scripts\activate # Windows

Step 3: Install dependencies

pip install -r requirements.txt

This installs Flask, PyTorch, TensorFlow, timm, and supporting libraries. Expect this to take a few minutes on a fresh install.

Step 4: Prepare a minimal dataset

Create a folder with at least two subfolders, one per category:

```
<project-root>/train/dataset/my_dataset/
+-- cats/
|   +-- photo1.jpg
|   +-- photo2.jpg
+-- dogs/
    +-- photo1.jpg
    +-- photo2.jpg
```

Aim for at least 50 images per category for meaningful results. The tool detects categories automatically from the folder names.

Step 5: Start the trainer

python app.py

You should see:

```
==============================================
  Model Training Web UI
  Open: http://localhost:5002
 ==============================================
```

Open http://localhost:5002 in your browser.

Step 6: Configure and launch a training run

1. Click "Train" in the navigation bar.
2. Give your run a name (e.g. "cats-vs-dogs-first-run").
3. Under Architecture, choose EfficientNet-B0 — a reliable default.
4. Set Epochs to 10 for a quick experiment.
5. Under Dataset Path, enter the path to your dataset folder
(e.g. <project-root>/train/dataset/my_dataset).
6. Click Start Training.

You are taken to the Monitor page, where you will see live log output and accuracy charts updating as each epoch completes.

Step 7: Download your model

Once training finishes, click "Models" in the navigation bar. Find your run and click "Download Package" to get a zip file containing the model weights, class names, and training history.

That is the complete workflow. The rest of this book explains what is happening at each step and how to get better results.

About the Publisher

Quiet Line Press creates simple, thoughtfully designed books and tools.

Visit:

https://quietlinepress.com

https://nextframe.dev

https://opennote.dev