WEEK 1 · DAY 1 · AFTERNOON

A bit of AI history and current AI Tech Stacks

My background is in Computer Vision.

A few years ago I submitted with some colleagues a “Trustworthy AI Systems” NSF/NRT grant

In my research I do AI/ML/Vision systems, including running a project where we deploy to NCMEC a tool to detect the hotel rooms in the background of sex trafficking images, which generates my interest in security of high stakes image ML tools.

https://pless.github.io/trustworthyAI/

1. Introduction to Trust & AI by Robert Pless
2. AI and High Tech Product Liabilty
3. AI, Publishing, and Ethics/Research Integrity
4. Mechanisms of Human Oversight in Training AI Systems
5. AI and Federal Food and Nutrition Programs
6. SOTA: AI Agents
7. Anthropological understanding of language; What is missing from Large Language Models (LLMs)
8. AI and Art Production

Project 1: Characterize AI Model Errors

Project 2: Games as a model of LLM alignment

Project 3: Selective Truthiness, Do LLMs tell different things to different people

Project 4: Personalizing Trust, Use AI for something that matters to you

Final Project: Write a paper submittable to AAAI AI Ethics and Society Conference

AI/ML was early on the ”replicability” train

Hugging Face

1998

MNIST

LeCun posts a handwriting dataset on his homepage.

2009 to 2012

ImageNet/AlexNet

An open dataset and a public contest. “AlexNet” in 2012 dominates the competition and shares complete specifications of their architecture.

2019

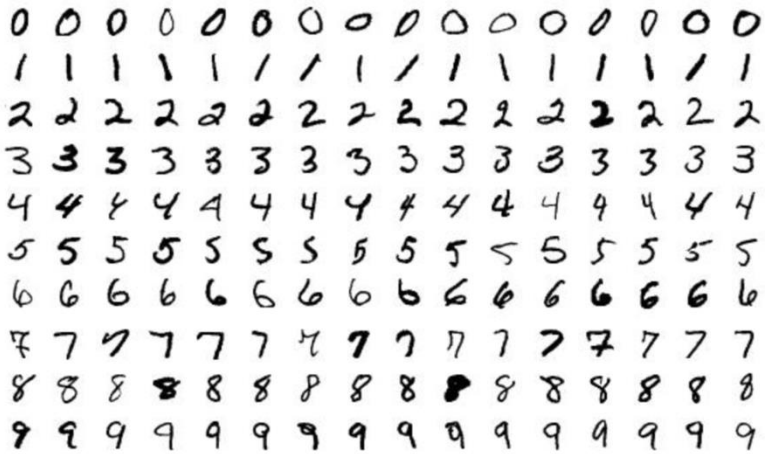
Reproducibility checklists

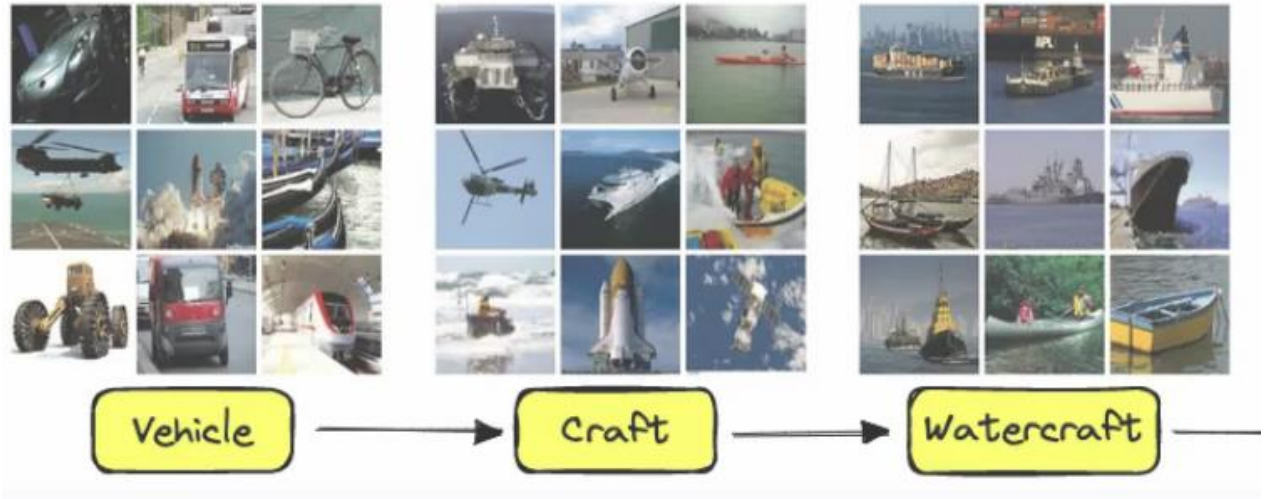
NeurIPS now requires a data checklist and code (& now an ethics statement). Papers sharing materials jump from about half to over three quarters.

Now

“Everything” on Hugging Face

Weights, datasets, and demos sit in one place. Sharing is the default, not the exception.

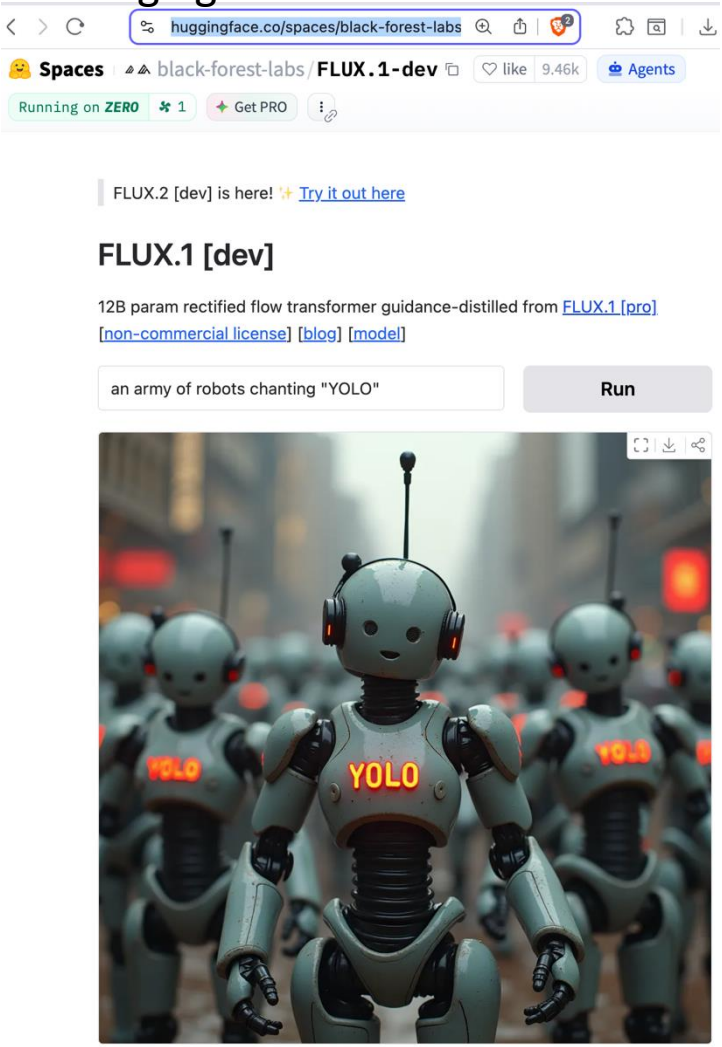




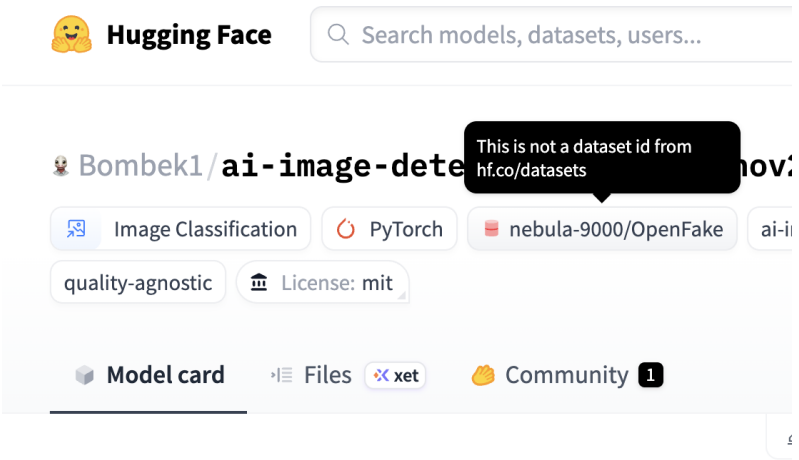
PART A · WHAT IS AVAILABLE

You can access/modify SOTA academic ML tools

Image generation



AI Image detection



Hugging Face

Afternoon goal:

Share some of the landscape of existing AI models and tools.

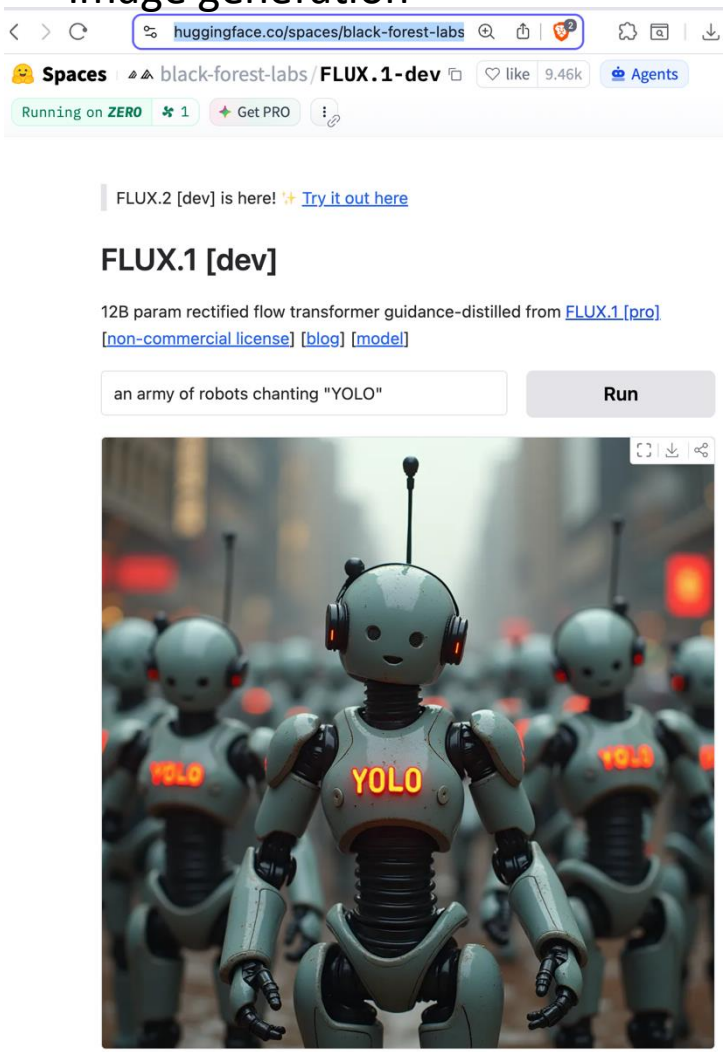
Give some examples of ways that you (or your students) can interact with these models.

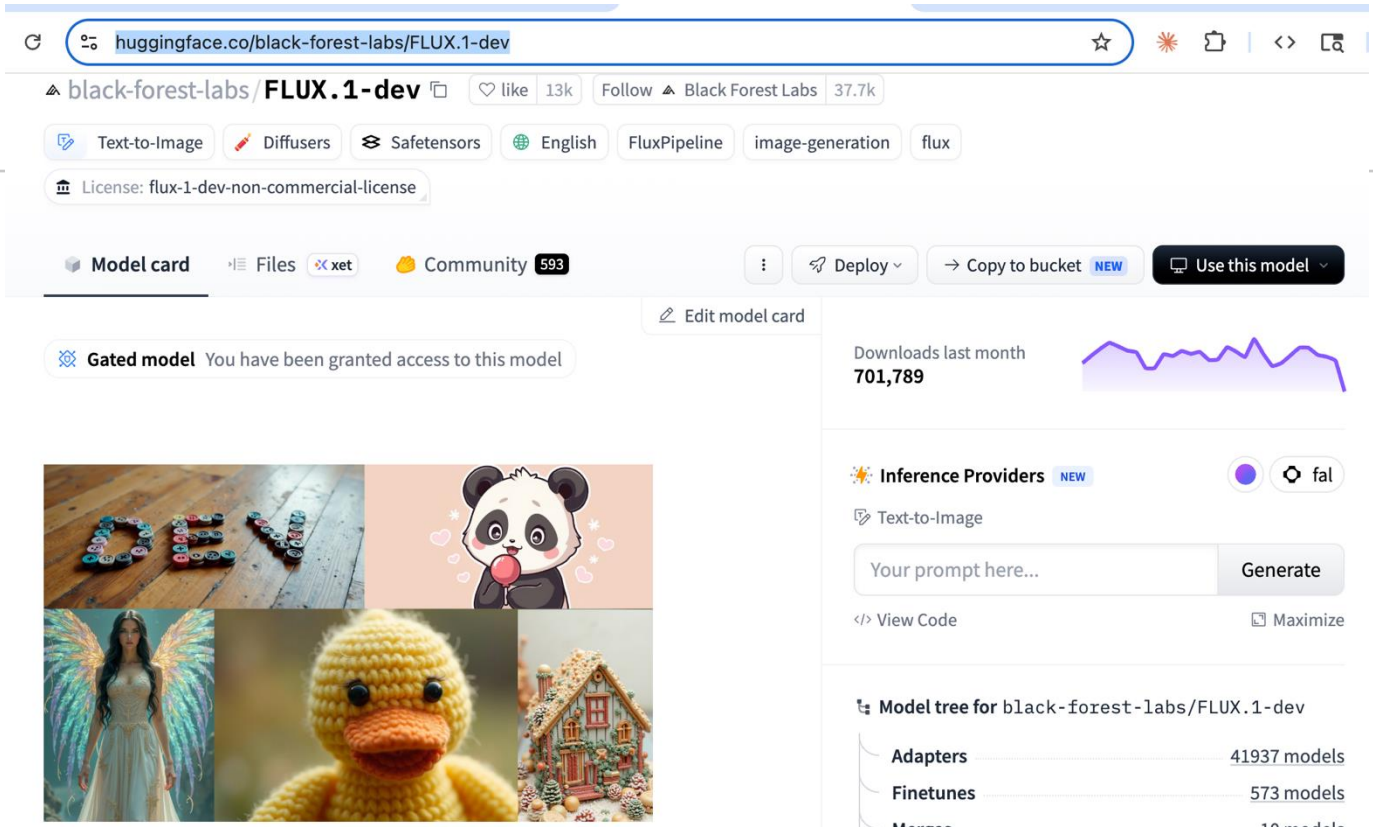
Explore the ”model zoo” in huggingface and share your favorite discoveries

PART B · WHAT IS AVAILABLE

https://huggingface.co/black-forest-labs/FLUX.1-dev

Image generation

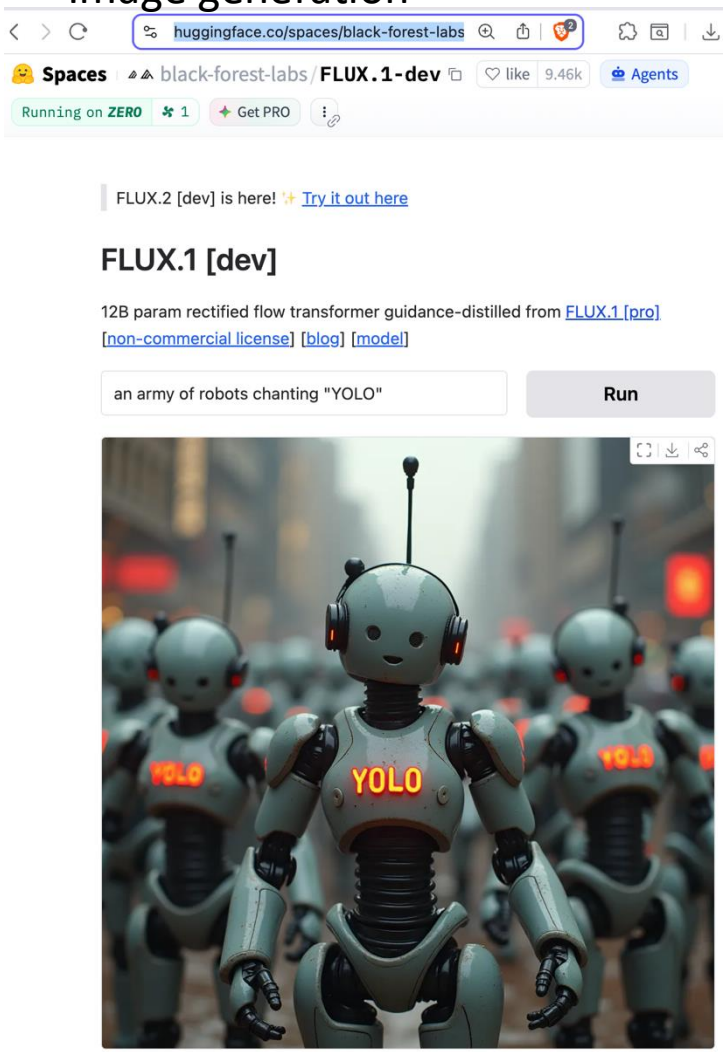


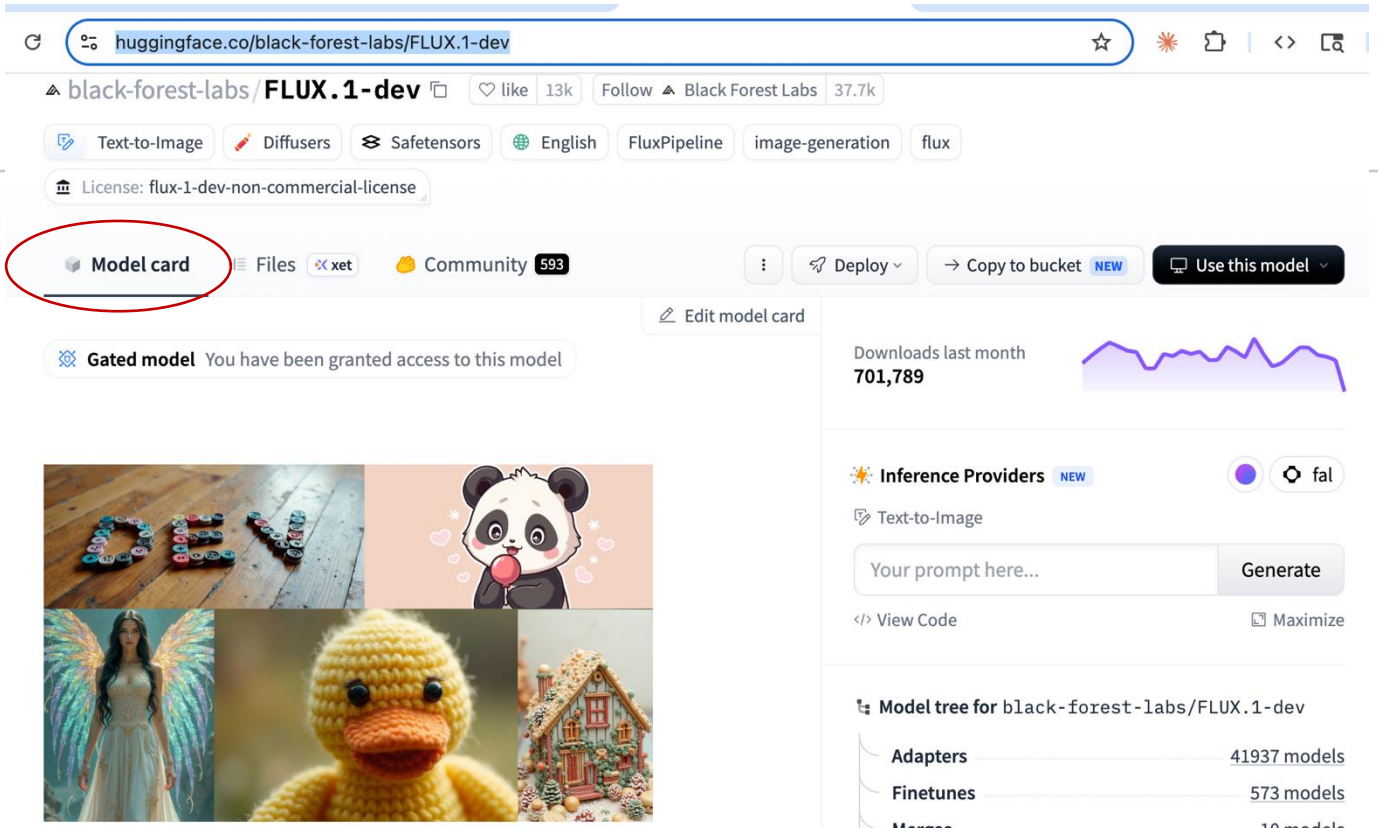


PART B · WHAT IS AVAILABLE

https://huggingface.co/black-forest-labs/FLUX.1-dev

Image generation

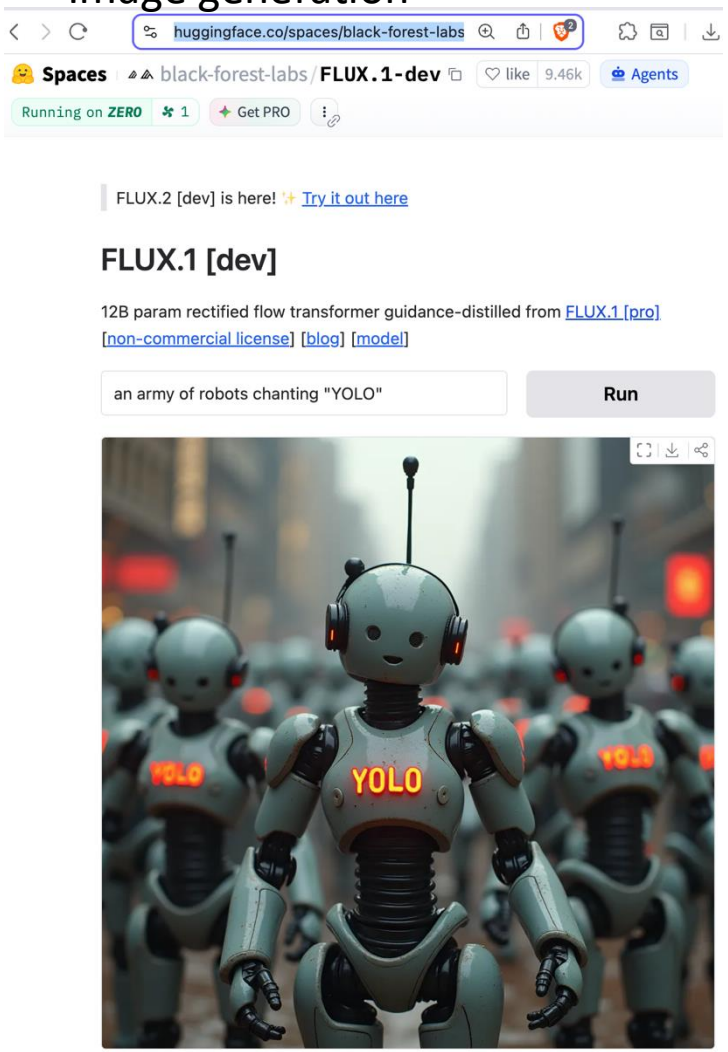


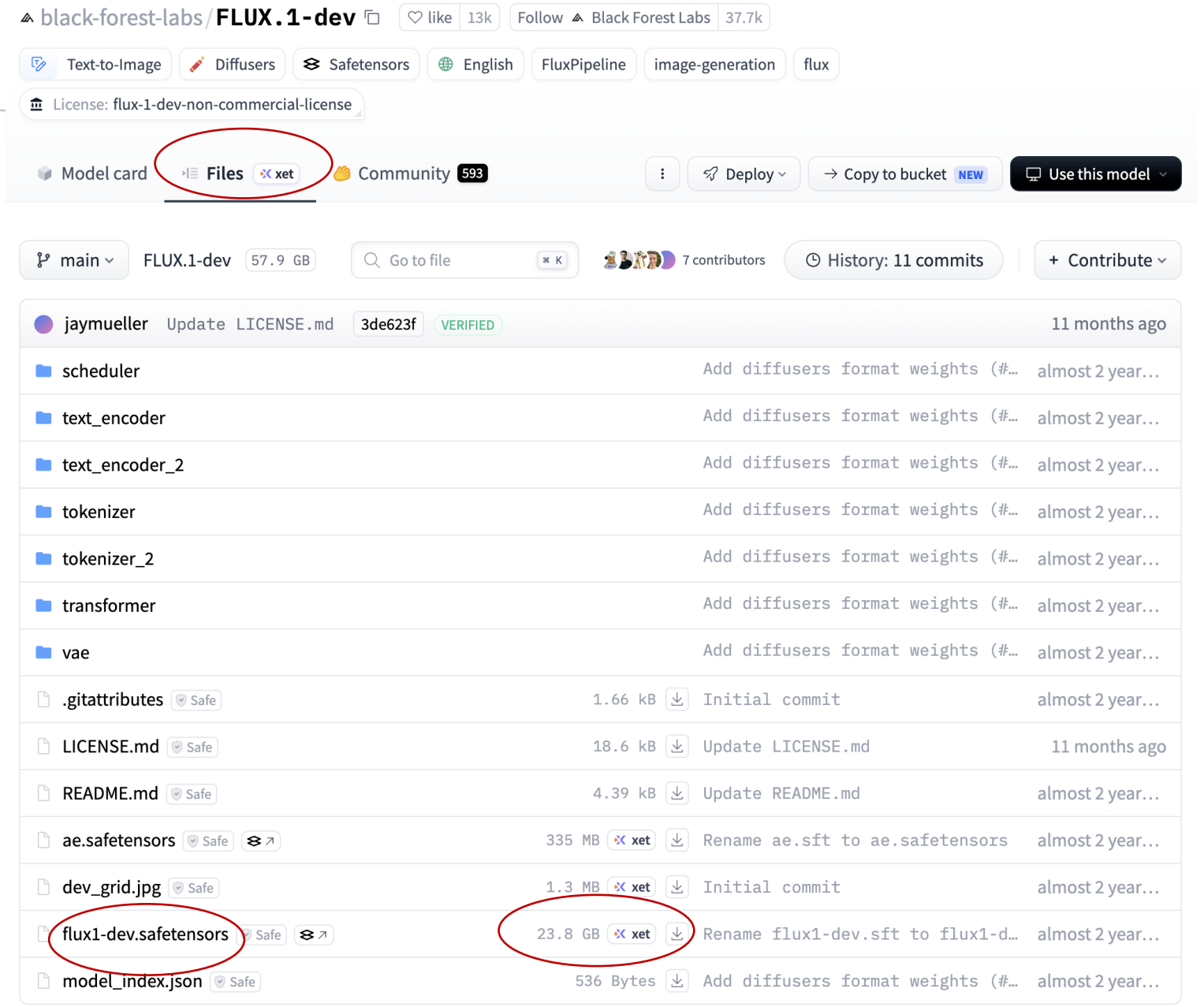


PART B · WHAT IS AVAILABLE

https://huggingface.co/black-forest-labs/FLUX.1-dev

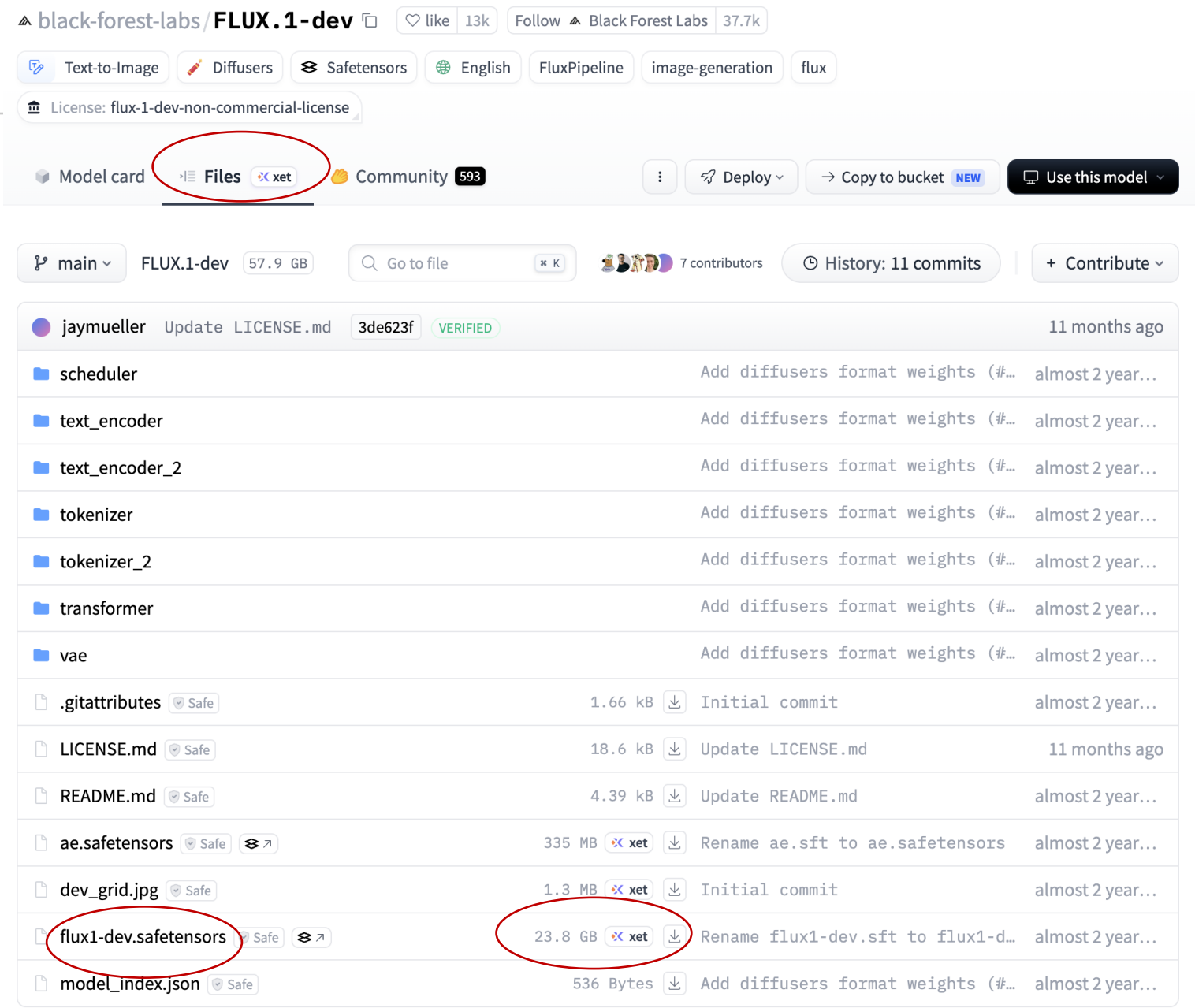
Image generation





https://huggingface.co/black-forest-labs/FLUX.1-dev

Image generation requires ~24 GB for inference (e.g. in order to produce an output), making it a laptop/server based inference model.



Inference Model Size

Even if you can download weights, how hard is it to run the inference?

Laptop

CPU, Apple Silicon, or a small GPU (about 8 GB)

BERT
text / NLP

YOLO
object detection

Whisper
speech to text

SAM
image segmentation

Stable Diffusion 1.5
text to image

GPU server

one 16 to 80 GB GPU (free Colab T4 at the entry)

FLUX.1
~24 GB native, ~8 to 13 GB quantized

Stable Diffusion XL / 3.5
text to image

Llama 8B to 70B
text / LLM

Data center

many GPUs in a cluster

Llama 405B
text / LLM

DeepSeek-scale (~671B)
text / LLM

The largest open models
frontier scale

PART B · WHAT IS AVAILABLE

“Open weights”

”Open”

Type	What it lets you do	Example models
Permissive	Use, modify, and ship almost anything, including commercial work. (MIT, Apache-2.0)	Whisper, SAM, Qwen, Mistral
Copyleft	Free to use, but derivatives and often network use must be shared back. (AGPL-3.0)	Some YOLO versions
Restricted community	Free with conditions: usage caps, banned uses, or no-training-other-models clauses.	Llama, FLUX.1-dev (non-commercial)

The first question is not "can I download it." It is "what does the license actually allow."

PART B · WHAT IS AVAILABLE

Closed models (for comparison)

Do not share their weights. Offered via API or as part of a device.

On-device

runs locally on your phone or laptop, but closed

Apple Intelligence
~3B on iPhone, iPad, Mac

Gemini Nano
Android AICore, Chrome

Phi-Silica
Copilot+ PCs

Mid tier (API)

the cheap, fast members of each lineup

Claude Haiku
fast, low cost

GPT-5 mini / nano
high volume

Gemini Flash / Flash-Lite
cheap, quick

Frontier (API)

served from large clusters, API only

Claude Opus 4.8
Anthropic

GPT-5.5
OpenAI

Gemini 3.1 Pro
Google

PART C · GETTING HANDS ON

How to reach each tier

Laptop

Install Ollama for local language models.
Use Python with scikit-learn or PyTorch.

GPU server

Use free Colab, rent a GPU from the Cloud or use university HPC.

Data center scale

Rent inference through a hosted API. There is no hardware to manage.
University systems/Hospitals sometime run their own model for data privacy reasons

PART C · GETTING HANDS ON

Anatomy of an API call

A hosted model is just an HTTP endpoint. You send a request describing what you want, and you get a a structured reply back.

```
# talk to a model over the network
from openai import OpenAI
client = OpenAI(api_key="sk-...")

resp = client.chat.completions.create(
    model="gpt-4o",   # the one line you'll swap next
    messages=[
        {"role": "system", "content": "You are a tutor."},
        {"role": "user",   "content": "What is a contrail?"},
    ],
)

print(resp.choices[0].message.content)
```

You send a request.

A **model name** plus a list of **messages**, each tagged with a role: `system`, `user`, or `assistant`. `assistant`.

You get structured JSON back.

The text lives at `choices[0].message.content`, alongside token counts and metadata.

Every hosted provider gives pretty much the same structure.

PART C · GETTING HANDS ON

OpenRouter: one key, many models

OpenRouter is the practical way to rent inference. One key reaches many hosted models, both open and closed.

```
# you change one string to swap models
model = "anthropic/claude-sonnet-4.6"
model = "meta-llama/llama-3.3-70b"
client.chat.completions.create(model, ...)
```

Routes your request across providers of a model finding ”best” (cheap, quick) response. Open models (Llama, DeepSeek, Qwen) have many companies that run data centers and serve the model to you.

Closed models (Claude, GPT) = there is only 1 provider. OpenRouter charges 5.5% fee.

Model price hugely variable across variables

Open model (hosted)	Input $/M	Output $/M
gpt-oss 20B	0.05	0.20
gpt-oss 120B	0.15	0.60
Llama-class 8B	0.18	0.18
Llama-class 70B	0.88	0.88
DeepSeek V4 Flash	0.14	0.28
Qwen 3.6 Plus	0.50	3.00
DeepSeek V4 Pro	2.10	4.40

Small open models undercut frontier APIs by 10–50× per token for routine tasks.

Output is the costly half

Generated tokens cost several times more than input tokens. Terse output saves money.

PART C · GETTING HANDS ON

Planning for a class: student keys and costs

- **Never paste keys in code.** Keys are credentials. Use Colab Secrets, and never commit them to git.
- **Set hard spend caps.** Put a budget limit on every account before the first call.
- **Start on free tiers.** Free credits cover a first lab. No card is needed to begin.
- **No student credit cards.** Provision keys for students. If they enter payment details they have high upside risk.

Class issues:

Where student data goes. Anything sent to a hosted API leaves your perimeter and may be logged. Student work and real logs can raise privacy and FERPA issues. The fix is to use synthetic data, or run an open model on your own hardware.

Models drift under you. A hosted model updates, and your assignment's expected output changes. A jailbreak you taught last week may get patched. The fix is to pin a specific open model for anything graded.

Goal for today

Where student data goes. Anything sent to a hosted API leaves your perimeter and may be logged. Student work and real logs can raise privacy and FERPA issues. The fix is to use synthetic data, or run an open model on your own hardware.

Models drift under you. A hosted model updates, and your assignment's expected output changes. A jailbreak you taught last week may get patched. The fix is to pin a specific open model for anything graded.

Survey:

Lab exercises: https://colab.research.google.com/drive/1_5hXaFWqS1hfkqGNX9dCWiSTo2tC0KG1