Methods of Model Train

1. DreamBooth

- **Goal:** Teach the model a specific person, object, or concept.
- **Input:** 4–30 images of a subject + class images.
- **Output:** Personalized model that can generate images using a trigger word.
- **Use Case:** Custom people, pets, objects, brands.
- **Tools:** Google Colab (TheLastBen, ShivamShrirao), Kohya-ss

---

2. LoRA (Low-Rank Adaptation)

- **Goal:** Fine-tune a model without changing the entire model weights.
- **Input:** 10–100+ images.
- **Output:** A lightweight.safetensors file that plugs into base model.
- **Use Case:** Styles, faces, cinematic tones, clothing, objects.
- **Tools:** Kohya-ss GUI, ComfyUI

---

3. Textual Inversion

- **Goal:** Teach the model a *word* (embedding) that represents a new concept.
- **Input:** 3–20 images + a unique placeholder token (e.g., <cinematic_guy>).
- **Output:** A .pt file (token embedding).
- **Use Case:** Simple styles, textures, poses, aesthetics.
- **Tools:** Automatic1111, Kohya GUI, Diffusers

---

4. Fine-Tuning (Full Model)

- **Goal:** Retrain both the UNet and text encoder for maximum model adaptation.
- **Input:** 1000s of images + GPU + time.
- **Output:** New full model.
- **Use Case:** Creating a new model for a specific domain or dataset.
- **Tools:** Huggingface Diffusers, Kohya-ss, DreamBooth + extra scripts

---

5. Full Model Training (From Scratch)

- **Goal:** Train a Stable Diffusion model from random weights.
- **Input:** Huge dataset (50k–500k images) + captions.
- **Output:** A new base model.

- **Use Case:** Research, commercial models (e.g., SDXL, RealisticVision).
- **Tools:** Custom PyTorch training scripts, Diffusers

---

6. Hypernetwork Training (older method)

- **Goal:** A modular training method to influence output without changing base model.
- **Status:** Largely replaced by LoRA.
- **Use Case:** Artistic or abstract style influence.

---

7. ControlNet Training

- **Goal:** Train models to respond to conditions like pose, depth, edge, etc.
- **Input:** Paired images and condition maps (e.g., pose images + normal photo).
- **Output:** A new ControlNet model file.
- **Use Case:** Cinematic scenes with exact poses, sketches, depth control.
- **Tools:** ControlNet training scripts (OpenPose, depth, canny, etc.)

---

8. Adapter Training / T2I Adapters

- **Goal:** Lightweight condition-based model tuning (similar to ControlNet, but smaller).
- **Use Case:** Depth, sketch, edge → image generation.
- **Status:** Experimental but effective for stylization and control.

---

9. Style LoRA / Layer-wise Fine-Tuning

- **Goal:** Train on *style* only using few images (e.g., "Christopher Nolan" style).
- **Input:** 20–100 images.
- **Use Case:** Cinematic look, lens effects, lighting tone.