THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Dog Breed Classification and Generation

Adam Stuhltrager | Juhua Deng | Sameer Batra | Sayan Patra

Table of Contents

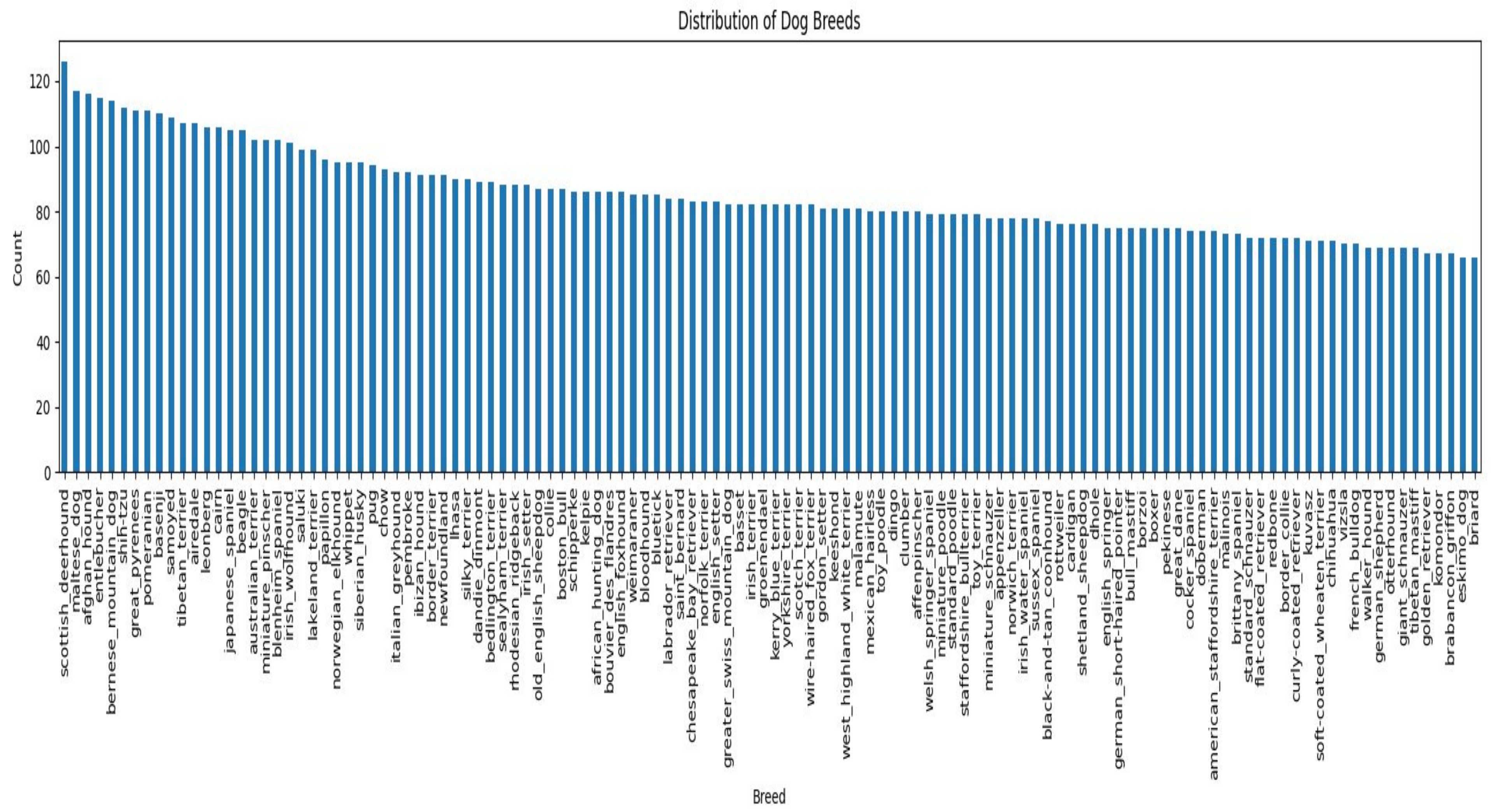
Project Goals

- **Develop a Highly Accurate Dog-Breed Classifier**
  Train and evaluate a ResNet-50 model capable of reliably distinguishing between **120 dog breeds**, even those with subtle visual differences.
- **Build a Photorealistic Dog Image Generator**
  Create an SDXL + LoRA–powered generator that can synthesize **anatomically consistent, breed-faithful** dog images in multiple visual styles.
- **Unify Classification and Generation into a Single System**
  Integrate the classifier and generator so the predicted breed can automatically guide or enhance image synthesis, forming a cohesive recognition-to-generation pipeline.
- **Ensure Strong Quantitative and Qualitative Performance**
  Optimize both components to achieve **high validation accuracy**, strong generalization, and visually realistic generation output across diverse breeds and poses.
- **Deploy an Interactive Application for Real-World Use**
  Build a Streamlit-based interface that allows users to **upload images, identify breeds, and generate custom dog images** in real time, demonstrating full system functionality.

Dataset Overview

Dog Breed Identification Dataset

- 10,222 Images
- Number of classes: 120 dog breeds
- Minimum Images per breed: ~60
- Maximum Images per breed: ~130



Distribution of Dog Breeds
Count
Breed

Classifier

Classifier

Data Preprocessing

- **Unified Image Format:**
  All inputs are converted to **RGB** to ensure consistent 3-channel structure, avoiding mismatches with the pretrained ResNet-50 expectations.
- **Spatial Normalization:**
  Images are resized to **224×224** followed by a **center crop** to remove spatial distortions and maintain uniform input size for convolution layers.
- **Tensor Conversion:**
  Images are transformed into **PyTorch tensors**, enabling batched GPU computation, gradient flow, and compatibility with the model pipeline.
- **Statistical Normalization:**
  Pixel values are standardized using **ImageNet mean/std** (μ=[0.485,0.456,0.406], σ=[0.229,0.224,0.225]) to align with pre trained feature distributions.
- **Transformation Consistency:**
  The *exact* validation preprocessing pipeline is replicated during inference to ensure **no distribution shift** between training and prediction.

Classifier

Data Loader Configuration

- **Batch Size Optimization:**
  Batch sizes of **128** balance memory efficiency and gradient stability, improving GPU utilization during training.
- **Randomized Training Batches:**
  Setting “`shuffle=True`” for training ensures each epoch sees a different sample order, enhancing generalization and preventing memorization.
- **Deterministic Evaluation:**
  Validation/testing Data Loaders keep “`shuffle=False`” to ensure **consistent, repeatable** evaluation metrics.
- **Parallel Data Loading:**
  Using **4–8 workers** loads multiple images simultaneously, reducing I/O bottlenecks and keeping GPUs fully utilized.
- **Pinned Memory Acceleration:**
  “`pin_memory=True`” speeds up CPU to GPU transfers by locking memory pages, enabling high-throughput batch delivery.

Classifier

Learning Rate Scheduler

Dynamic LR Adjustment:
Schedulers modify the learning rate during training to allow **fast initial learning** and **stable fine-tuning** as gradients get smaller.
Avoiding Optimization Pitfalls:
LR decay prevents the optimizer from overshooting minima or oscillating, especially important for deep networks like ResNet-50.
Scheduler Options:

- **Step LR:** periodic LR drops
- **CosineAnnealing:** smooth cyclic decay
- **ReduceLROnPlateau:** triggered when validation loss stagnates

Better Convergence Quality:
Lower LR in later epochs enables the model to fine-tune high-level features, improving classification accuracy on similar dog breeds.
Training Stability:
Controlled LR decay reduces gradient explosion risk and ensures smoother loss curves throughout training.

Classifier

Early Stopping

- **Continuous Validation Monitoring:**
  The system tracks **validation loss each epoch**, acting as a signal of generalization performance.
- **Patience-Based Stopping:**
  Training halts when the model fails to improve for a specified number of epochs, preventing wasteful computation.
- **Overfitting Prevention:**
  Early stopping stops learning before training noise is memorized, enhancing real-world predictive robustness.
- **Best Checkpoint Preservation:**
  The checkpoint with the lowest validation loss is retained, ensuring deployment uses the **optimal** model version.
- **Efficient Training Time:**
  Cuts off unnecessary training epochs, especially when improvements plateau, enabling faster experimentation cycles.

Classifier

Integrated Training Workflow

- **Input Standardization:**
  Preprocessing ensures images match the input distribution expected by pretrained ResNet-50, improving feature extraction fidelity.
- **Data Feeding Efficiency:**
  An optimized Data Loader pipeline ensures smooth, non-blocking GPU usage, maintaining high training throughput.
- **Adaptive Learning Control:**
  Learning rate schedulers dynamically adjust optimization behavior, stabilizing convergence and improving accuracy.
- **Generalization Safeguards:**
  Early stopping protects against overfitting and ensures the best-performing model is always selected.
- **Robust Final Model:**
  Together, these components form a consistent, efficient, and high-performing training loop for dog-breed classification.

Classifier

Model Architecture

- **Backbone: ResNet-50 Deep CNN**
  Utilizes a 50-layer Residual Network consisting of convolutional layers with skip connections, enabling stable training of deep architectures by mitigating vanishing gradients.
- **Bottleneck Residual Blocks**
  Each block uses a **1×1** → **3×3** → **1×1** convolution structure, reducing computational cost while allowing extraction of fine-grained texture patterns crucial for distinguishing similar dog breeds.
- **Global Average Pooling Layer**
  Compresses spatial feature maps into a single vector per channel, making the network robust to object location and improving breed classification accuracy.
- **Custom Fully Connected Classifier Head**
  The original ImageNet FC layer is replaced with a **Linear(num_features** → **num_classes)** layer aligned with the exact number of dog breeds, enabling fine-tuned classification.
- **Softmax Output & Confidence Scoring**
  Final logits are passed through a softmax layer to produce breed-wise probabilities, allowing extraction of both **top-1 prediction** and **confidence score** for deployment and UI integration.

Classifier

Results

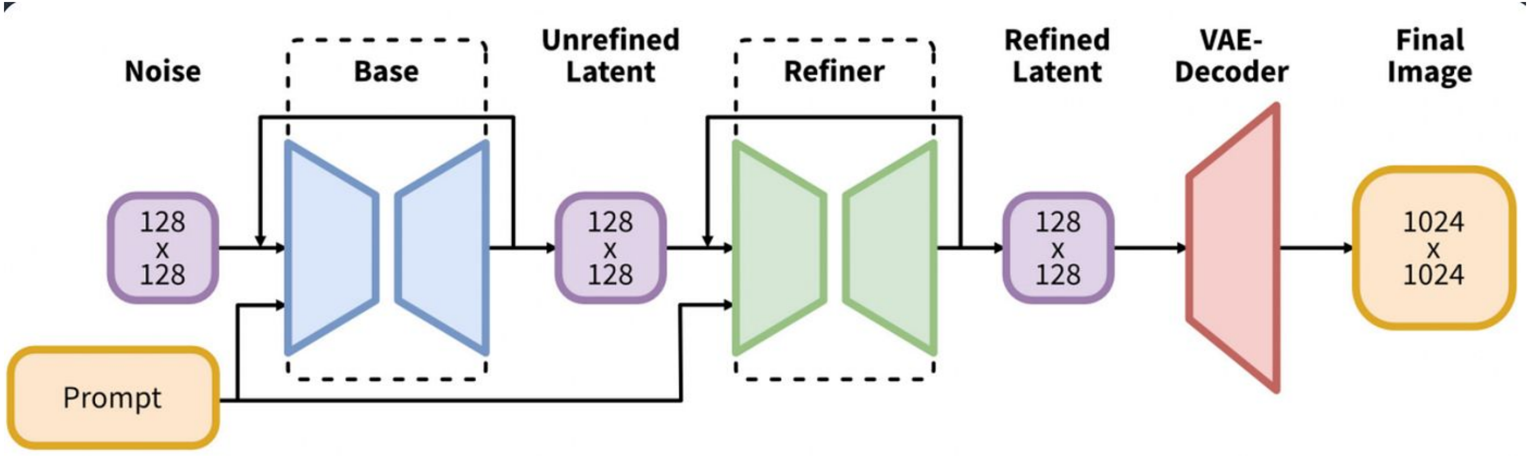
	Macro	Micro	Weighted
Precision	0.8508	0.8508	0.8601
Recall	0.8461	0.8508	0.8508
F1-score	0.8435	0.8508	0.8508
Accuracy			0.8508

Generator

Generator

SDXL

Using a large dual-UNet diffusion architecture that gradually denoises latent representations into high-resolution images. It combines two powerful text encoders to understand prompts more precisely, and its two-stage generation process (Base + Refiner) allows the model to capture global structure first and then add fine-grained details.



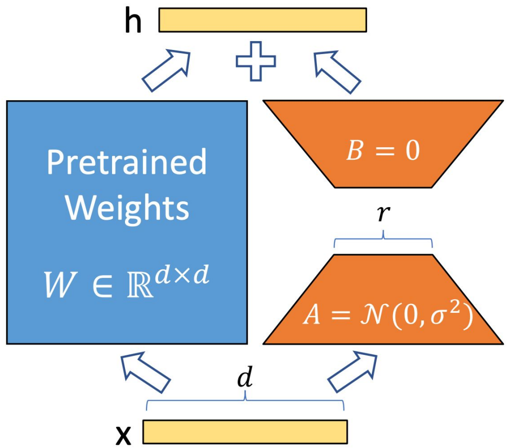
Generator

	SDXL	SD 1.5	SD 2.1
Original Resolution	1024x1024	512x512	768x768
Parameters	2.6B	0.86B	0.86B
Visual Quality	Significantly higher fidelity	Moderate	Improved over 1.5
Prompt Adherence	Excellent	Moderate	Good

Generator

LoRA

An efficient fine-tuning method that modifies a large model by adding a small low-rank update to its original weight matrices. Instead of training the full weight W, LoRA freezes W and learns an update represented as



Generator

Model Loading & Initialization

- **Base Model Download:**
  Loads SDXL via `StableDiffusionXLPipeline.from_pretrained()` using either fp16 (GPU) or fp32 (CPU).
- **torch_dtype Selection:**
  Dynamically sets dtype to **float16** on CUDA for speed and memory efficiency; float32 on CPU for compatibility.
- **LoRA Repository Handling:**
  LoRA repo & weight names are configurable:
  `lora_repo`, `lora_weight_name`, `lora_scale`.
- **Style Customization:**
  LoRA scaling enables mixing SDXL's base model with stylistic layers (e.g., manga, cinematic, line-art).

Generator

Prompt Engineering

- **Primary Prompt (Limit to 77 tokens):**
  Accepts breed-specific text and adds detailed descriptors such as **anatomy**, **pose**, **style**, **linework**, etc.
- **Negative Prompt:**
  Carefully crafted to prevent common SDXL artifacts:
  extra tails, extra limbs, distorted anatomy, CGI look, blurred areas, missing legs.
- **Dog Anatomy Optimization:**
  Includes constraints like *“single visible tail”*, *“proper proportions”*, *“natural limb spacing”* to avoid morphological hallucinations.
- **Style Control:**
  Can be switched between Manga/Anime/Pastel Anime/Pixel Art.

Generator

Negative Examples:

Multiple Tails

Extra Limbs

Generator

Settings

- **Seeded Generator:**
  Uses `torch.Generator` to enforce deterministic output when a `seed` is provided.
- **Additional Cross-Attention Parameters:**
  If LoRA is active, the generator injects `"cross_attention_kwargs": {"scale": lora_scale}` to blend styles smoothly.
- **Sampling Configuration:**
  - Resolution: **1080 × 720**
  - Inference Steps: **50**
  - Guidance Scale: **5.0**

    These balance quality, consistency, and style strength.
- **Pipeline Call:**
  Executes `self.pipe()` with prompt, negative prompt, shapes, guidance, generator, and LoRA scaling.

Generator

Realistic SDXL Generation Overview

- **Text-to-Latent Pipeline Creation**
  The system constructs a two-stage SDXL pipeline (Base → Refiner), enabling structured latent formation first, followed by refinement, mimicking how real image formation happens incrementally.
- **Anatomy-Grounded Prompt Injection**
  Prompts are dynamically built from breed-specific anatomical descriptors and global biomechanical priors, ensuring that SDXL starts with correctly conditioned structural information before sampling begins.
- **Negative Prompt Enforcement**
  Combined negative prompts strictly penalize SDXL's common anatomical failure modes (extra limbs, fused paws, warped joints), forcing the pipeline to remain anatomically aligned during diffusion steps.
- **Sequential Denoising (Base → Refiner) Workflow**
  The Base model generates the global structure in latent space, while the Refiner model sharpens texture, fur detail, paw edges, and lighting, functioning as a second "micro-detail" denoising phase.
- **Real-Life Post-Processing Integration**
  After SDXL generates the refined image, custom physical post-processing (microfur, tone curve, sensor noise) is applied to align the final output with real camera characteristics.

Generator

Prompt Engineering Pipeline

- **Breed-Specific Anatomy Extraction**
  The system retrieves skull, fur, paws, tail, and body descriptions from BREED_ANATOMY and merges them with hind-leg biomechanics for structurally correct prompt conditioning.
- **Generic Prior Fallback Mechanism**
  When the breed is not found, the pipeline replaces missing details with a robust global canine anatomy prior, ensuring SDXL always receives sufficiently detailed geometric constraints.
- **Photographic Prior Injection (PHOTO_PRIOR)**
  Every prompt is reinforced with DSLR-specific cues—lens focal length, RAW texture behavior, dynamic range—which trains SDXL to hallucinate a *photograph*, not a stylized render.
- **Pose Conditioning and Leg Visibility**
  Full-body side-view constraints ensure that SDXL must render all four limbs, distribute weight naturally, and expose the hind-leg structure — a common failure area corrected here.
- **Unified Prompt Chain for Base and Refiner**
  Both SDXL stages share the same positive/negative prompt embeddings, ensuring consistent structural intent across denoising phases.

Generator

Sampling & Latent Pipeline

- **Timesteps and Scheduler Configuration**
  The Euler Ancestral scheduler creates a stable noise-to-latent trajectory, balancing sharp detail with anatomical accuracy across the Base and Refiner phases.
- **Denoising Split Strategy**
  Total steps are divided between the Base (structure) and Refiner (detail). The Base denoises until `denoising_end`, handing the latents to the Refiner at the correct "noise depth."
- **Prompt Embedding Encoding**
  Both positive and negative prompts are encoded into high-dimensional embeddings. SDXL uses these embeddings as conditioning at every timestep through cross-attention.
- **Latent Representation Stabilization**
  Intermediate latent images are generated in a noise-controlled latent space, enabling SDXL to maintain consistent anatomy before translating them into pixel-space.
- **Refiner Enhancement Pipeline**
  The refiner continues denoising only the last portion of the pipeline, sharpening hair strands, eyes, paw textures, and structural edges without altering global anatomy.

Generator

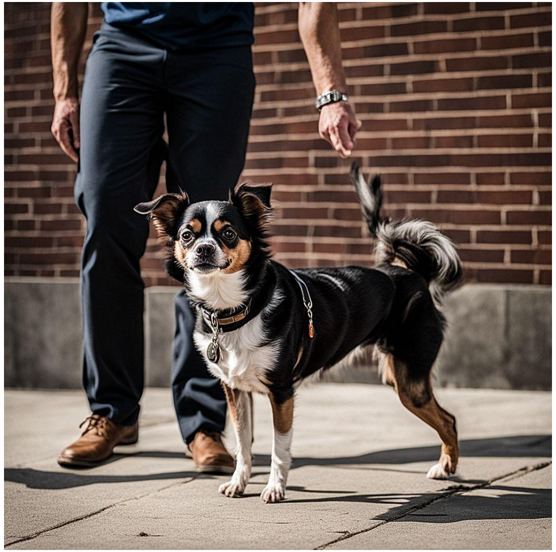
Anatomy-Control Pipeline

- **Hind-Leg Biomechanics Enforcement**
  The generator inserts detailed biomechanical descriptors for femur–tibia proportions, stifle angles, and hock behavior to force SDXL into anatomically valid limb representation.
- **Breed-Specific Additive Anatomy**
  Skull shape, paw types, coat density, tail carriage, ear posture, and gait descriptions are appended to the prompt, giving SDXL explicit structural targets.
- **Global Canine Anatomy Stack**
  Even if breed-specific details fail, the pipeline always includes the general canine skeletal and muscular structure, preventing SDXL from drifting into iconographic or stylized forms.
- **Hard Negative Penalties for Limb Failures**
  Negative prompts explicitly describe unacceptable outputs (fused paws, extra legs, misplaced joints), shaping the loss landscape to punish structural hallucinations.
- **Prompt Embedding Uniformity Across SDXL Stages**
  Ensures anatomy constraints affect both the early low-frequency structure (Base) and final micro-detail refinement (Refiner), maintaining consistency throughout the pipeline.

Generator

Negative Examples

Incorrect perspective, multiple limbs



Lack of limb

Deformed Face

Generator

```
Shoulder
   |
   |  Upper arm (humerus)
   |
   o  Elbow
   |
   |  Forearm (radius/ulna)
   |
   o  Carpus (wrist) ← bends forward
   |
   |  Pastern
   |
  🐾 Paw
```

Anatomy Map

```
Hip
 |
 |  Femur (thigh)
 |
 o  Stifle (knee) ← bends forward
 |
 |  Tibia (shin)
 |
 o  Hock (ankle) ← bends backward, elevated
 |
 |  Metatarsus (rear pastern)
 |
🐾 Paw
```

Generator

- The negative prompt modifies the noise prediction:
- $\epsilon guided=\epsilon uncond+s\cdot(\epsilon cond-\epsilon uncond)-sneg\cdot(\epsilon neg-\epsilon uncond)$
- **Without negative prompt:**
- $\epsilon=\epsilon uncond+s\cdot(\epsilon pos-\epsilon uncond)$
- **With negative prompt:**
- $\epsilon=\epsilon uncond+s\cdot(\epsilon pos-\epsilon uncond)-sneg\cdot(\epsilon neg-\epsilon uncond)$
- Where:
- ϵpos = prediction guided by "golden retriever puppy"
- ϵneg = prediction guided by "cartoon, extra legs, blurry..."
- sneg= negative guidance scale (usually same as ss)
- **Effect:** Pushes the generation away from the negative concepts while pulling toward positive ones.

Generator

Photoreal Post-Processing Pipeline

Microfur Luminance Enhancement
Operates only on the Y-channel of YUV space, enhancing natural fur texture while avoiding color distortion, mimicking real sensor fur granularity.
Real Camera Noise Simulation
Shot-noise and chroma-noise models recreate real sensor imperfections, removing the overly smooth “diffusion look” and increasing perceived realism.
Real Tone Curve Application
Applies DSLR-like S-curve contrast, gentle gamma shift, and dynamic range shaping to approximate professional camera output.
Detail Preservation Without Oversharpening
Adjusts microstructure without edge halos, maintaining real photographic softness while preserving fine detail such as paw edges and muzzle texture.
Final Natural-Looking Composition
The combined post-processing stages align the image with real-world photographic expectations, making the output visually indistinguishable from actual dog photos.

Generator

Seed-Controlled Determinism

- **Seed-Controlled Determinism**
  A per-device `torch.Generator` ensures exact reproducibility — same seed, same anatomy, same lighting.
- **Snap-to-8 Resolution Handling**
  Automatically adjusts width & height to multiples of 8 for SDXL compatibility, preventing tensor misalignment.
- **Automated CLI-Based Execution**
  The pipeline supports arguments (`--breed`, `--lora`, `--seed`, `--out`) enabling automated batch generation.
- **Timestamped Image Saving**
  Output file names include timestamps + breed name for dataset consistency and experiment tracking.
- **Production-Ready Modular Design**
  Pipeline is fully modular: anatomy engine, prompt builder, sampler, post-processor, and file saver operate as independent layers for easy extension.

Findings

- **High-Accuracy Dog Breed Classification Across 120 Classes**
  Achieved **~0.85 accuracy and F1-macro**, demonstrating strong generalization even on visually similar breeds.
- **Realistic, Anatomy-Aware Image Synthesis for 120 Breeds**
  The SDXL + LoRA generator produces coherent, breed-faithful images across multiple visual styles with controlled anatomy and pose.
- **Robust Performance Validated with Multiple Metrics**
  Consistent macro/micro/weighted F1 scores confirm balanced predictions across both rare and common breeds.
- **Unified Recognition–Generation Pipeline**
  Successfully combined ResNet-50 classification with SDXL image generation into a single automated system for end-to-end breed understanding.
- **Interactive Deployment via Streamlit Application**
  Built a full-stack interface enabling real-time breed prediction and customizable dog-image generation for user exploration.

Conclusion

- **Unified Discriminative + Generative System**
  This project successfully integrates a **ResNet-50 breed classifier** with an **SDXL + LoRA image generator**, demonstrating how recognition and synthesis can operate together in a single interactive workflow.
- **Strong Classification Performance**
  The classifier achieves **~0.85 macro, micro, and weighted F1**, showing balanced performance across both frequent and rare breeds. Stable micro-accuracy (0.8508) confirms consistent prediction quality on the full 120-breed dataset.
- **Multi-Style, Anatomy-Aware Image Generation**
  The SDXL pipeline produces coherent dog images across **five distinct styles**, with LoRA providing controllable stylization. Despite diffusion natural structural challenges, the system maintains breed identity and visual consistency.
- **Identified Limitations**
  Current constraints include:
  - Classifier restricted to **closed-set 120 breeds**
  - Occasional SDXL **anatomical inconsistencies** in limbs and pose
  - High GPU memory and compute cost during generation
    These highlight clear opportunities for refinement.
- **Future Impact and Extensions**
  The project lays a strong foundation for advancements such as **open-set breed classification**, improved **anatomical constraint modules** in diffusion models, cross-breed interpolation, and optimized deployment for real-time applications.

Streamlit

Link(R): http://98.83.136.175:8888/

Link: http://54.162.224.22:8888/

DEMO

Thank you