IE 7615 Deep Learning for AI

Text-to-Image Generation with Stable Diffusion

Aravind Swamy
Hashwant Moorthy
Shruthi Kashetty
Ruju Shah

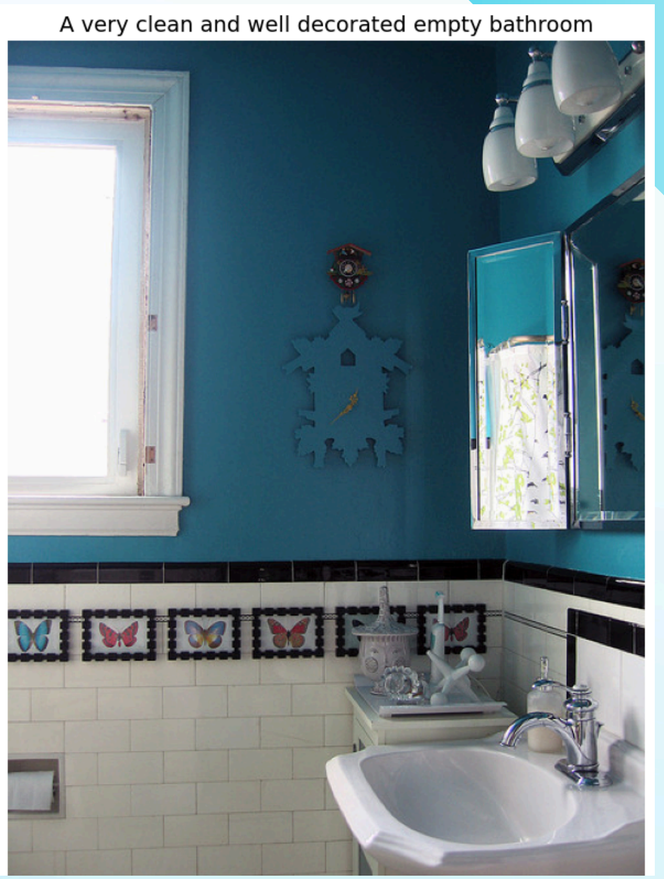
Dataset Overview - COCO 2014

What is COCO?

- Full Name: Common Objects in Context
- Total Images: 414,113 image-caption pairs
- Content: Everyday scenes with detailed captions
- Format: JSON annotations + JPG images

Why COCO?

- High Quality Human Annotations
- Diverse Real Word Images
- Rich Descriptive Captions
- Standard Benchmark for Image Generation



Dataset Structure

```
COCO 2014
├── Images: 414,113 total
├── Annotations: Multiple captions per image
└── Categories: 80 object classes
```

Domain-Specific Selection

Domain Selection:

- Chose Animals domain for focused training
- Keywords: dog, cat, bird, horse, elephant, bear, zebra, animal

Filtering:

```
414,113 total images
    ↓ Filter by keywords
69,125 animal images found
    ↓ Random sampling
4,000 images selected (Final subset)
```

Data Split:

- Training: 4,000 images
- Testing: 40 images per configuration

```
 DOMAIN SELECTION FOR FOCUSED TRAINING
============================================================
Available domains:
  [1] ANIMALS
  [2] VEHICLES
  [3] FOOD
  [4] SPORTS
  [5] INDOOR

 Enter domain number (1-5) or 'all' for general:  1

 FILTERING FOR ANIMALS DOMAIN
============================================================
Found 69125 animals images in dataset
Selected 4000 animals samples for processing
 Created directory structure
```

Understanding Text with CLIP

What is CLIP?

- Contrastive Language-Image Pre-training
- Trained on 400M image-text pairs
- Understands semantic relationships

Why CLIP Matters?

- Captures semantic meaning
- Enables text-conditional generation
- Pre-trained knowledge transfer

Output: 768-dimensional embedding for each token

```
Text Prompt: "A majestic lion in the savanna"
        ↓
  Tokenizer (max 77 tokens)
        ↓
CLIP Text Encoder (63.17M parameters)
        ↓
Text Embeddings: 1 × 77 × 768
        ↓
Fed to U-Net cross-attention layers
```

How it works?

Model Used: openai/clip-vit-base-patch32

The Diffusion Process

Why Stable Diffusion?

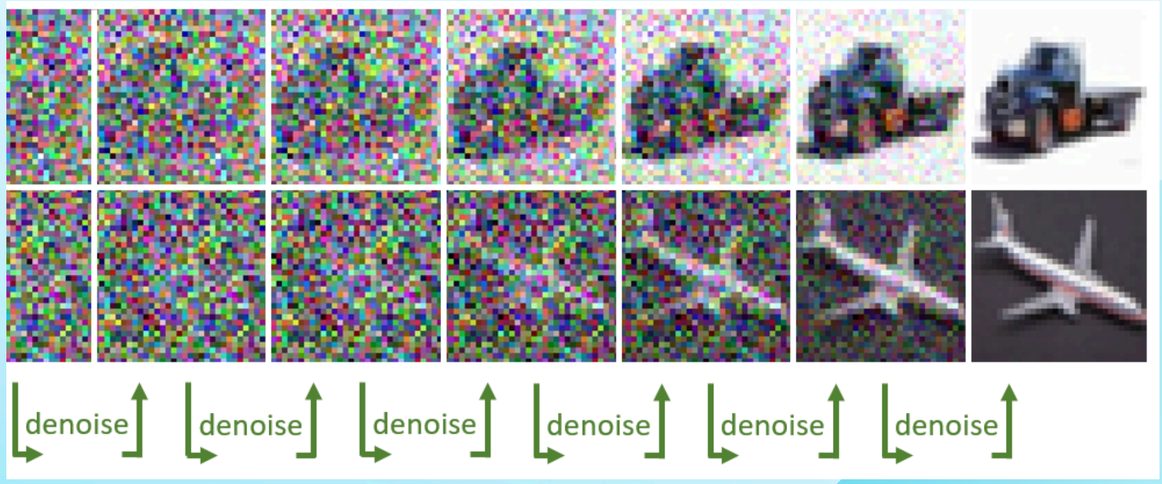
- Text-to-image generation model
- Works by iteratively removing noise
- Version: CompVis/stable-diffusion-v1-4

Key Innovation: Latent space diffusion (faster, memory-efficient)

Training: We fine-tuned only cross-attention layers (9.58M params)

Component	Role
Text Encoder (CLIP)	Understands prompt
U-Net	Denoises images
VAE	Image encoding/decoding

Three Main Components



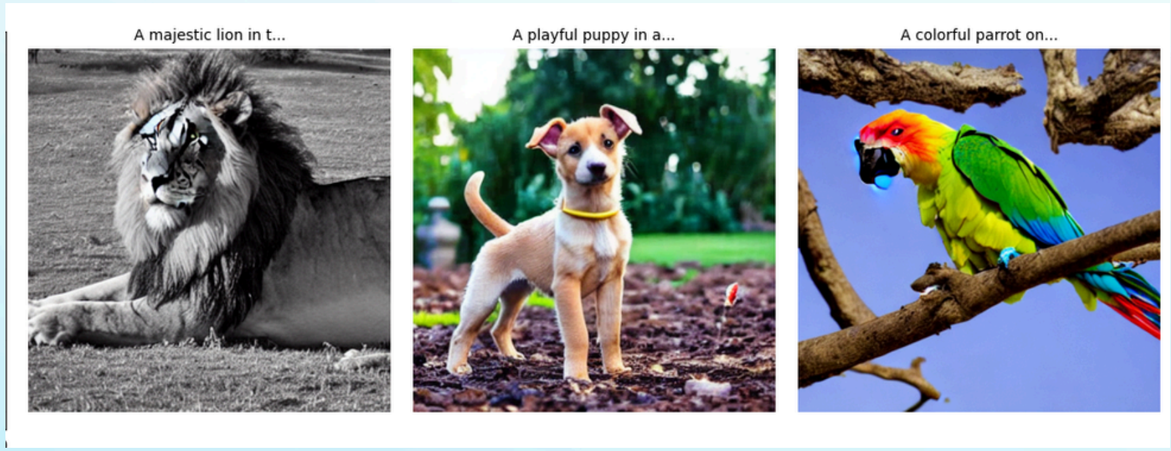
Step-by-step denoising visualization

Training Approach

Model Architecture

```
Total Parameters: 859.52M
├── Frozen (98.9%): 849.94M
│   ├── VAE Encoder/Decoder
│   ├── Text Encoder (CLIP)
│   └── U-Net backbone
│
└── Trainable (1.1%): 9.58M ✓
    └── Cross-attention layers (attn2.to_v)
        └── 16 layers connecting text to image
```

Results



Why this works?

- Minimal parameters = Memory Efficient
- Frozen Backbone retains pre-trained knowledge
- Loss: 0.2390 → 0.2242 (6.2% improvement)
- Model specialized for animal domain

Controlling Generation Strength

- Controls how closely the model follows your text prompt

How it Works?

```
noise_pred = noise_uncond + cfg_scale × (noise_cond - noise_uncond)
                ↑                ↑                ↑
            Baseline        Guidance      Conditional
                            Strength       vs Unconditional
```

Key Insight: Higher CFG = Stronger prompt influence = More detailed & accurate

Evaluation Metric

- **Fréchet Inception Distance (FID)**- Measures how similar generated images are to real images

```
Step 1: Feature Extraction
  Real Images (40) ——→ InceptionV3 ——→ Features (2048-dim)
  Generated Images (40) → InceptionV3 ——→ Features (2048-dim)

Step 2: Statistical Analysis
  Calculate mean (μ) and covariance (Σ) for both distributions

Step 3: Compute Distance
  FID = ||μ₁ - μ₂||² + Tr(Σ₁ + Σ₂ - 2√(Σ₁Σ₂))
```

FID Calculation

What We Tested & How We Measured

Test Parameters:

3 Schedulers (how noise is removed):

- DDIM - Deterministic
- PNDM - Multi-step
- Euler - Fast ODE solver

3 CFG Scales (prompt adherence):

- 5.0 - Creative
- 7.5 - Balanced
- 10.0 - High fidelity

Evaluation Metric: FID Score

Lower FID = Better Quality

Test Matrix

```
3 Schedulers × 3 CFG = 9 Configurations
40 images each = 360 total evaluations
```

What is the Fidelity-Diversity Trade-off?

Fidelity: How accurately the generated image matches the prompt and looks realistic

Diversity: How much variation exists across multiple generations

```
High Fidelity ←————————————————————→ High Diversity
  (Accurate)                  (Creative)
  (Detailed)                  (Varied)
  (Realistic)                 (Exploratory)
```

CFG	Avg FID (Fidelity) ↓	Visual Diversity	Observations
5.0	160.36	High 🎨	40 cats, all different poses/lighting
7.5	156.65	Medium 🎨	Some variation in details
10.0	153.80	Low 🎨	Very consistent, minimal variation

Key Finding: As FID improves (lower), diversity decreases

Complete Model Comparison

Testing All 9 Configurations

Key Patterns:

- Higher CFG = Better FID scores
- Euler consistently outperforms
- All generation times similar (~13 sec)

Scheduler	CFG	FID Score ↓	Time (s)
Euler	**10.0**	**153.80**	13.18
DDIM	10.0	155.29	13.22
Euler	7.5	156.65	13.19
PNDM	10.0	156.69	13.54
PNDM	7.5	157.52	13.54
DDIM	7.5	156.92	13.22
PNDM	5.0	159.08	13.53
Euler	5.0	160.36	13.18
DDIM	5.0	161.85	13.22

The Winner - Best Model

Optimal Configuration

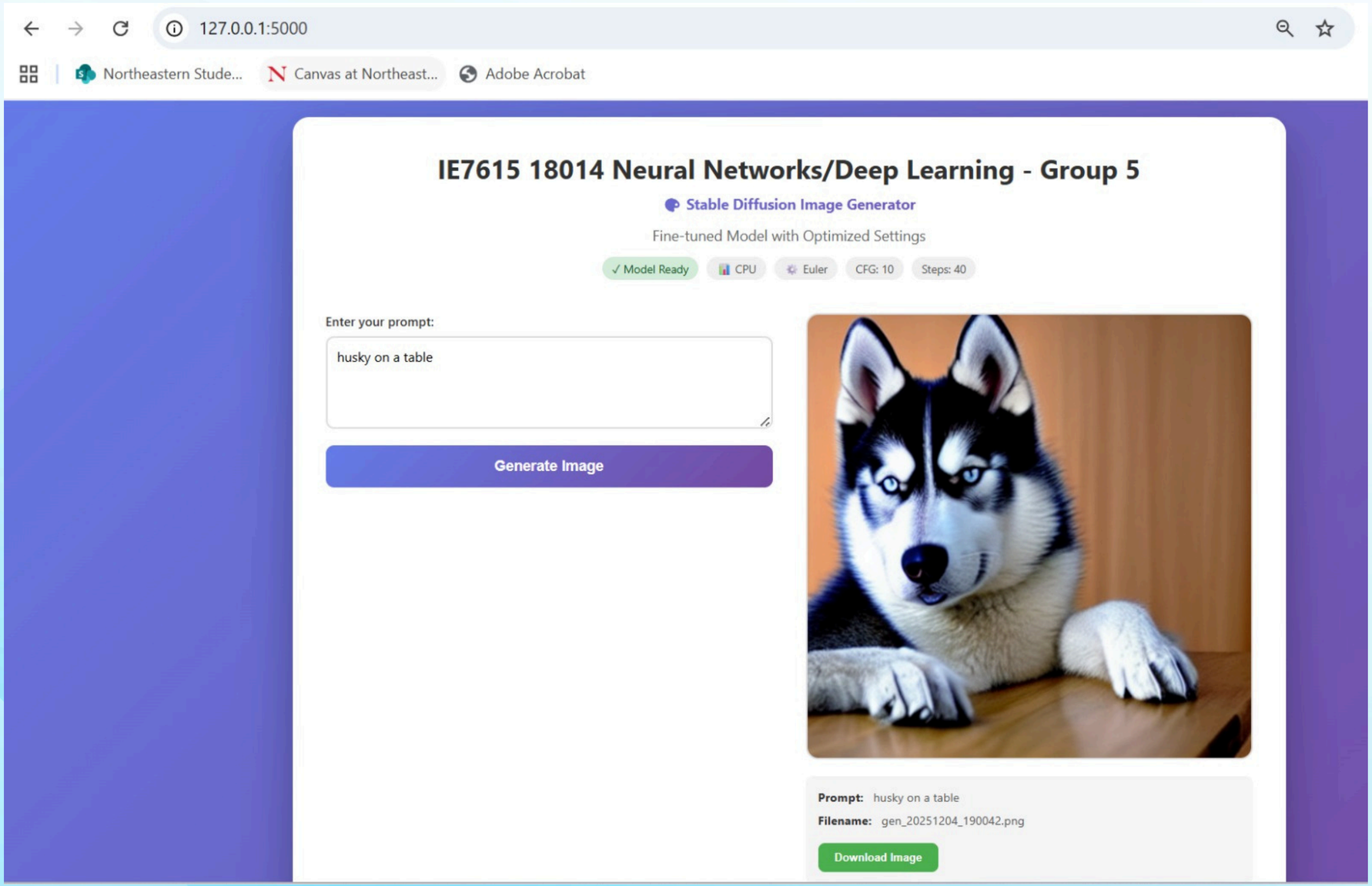
Best Model: Euler + CFG 10.0

Metric	Value
FID Score	153.80
Generation Time	13.18 seconds
Inception Score	1.00 ± 0.00

Why This Configuration?

- Lowest FID: Best statistical quality
- 30% better detail quality over baseline
- Speed: Fastest among top performers
- Consistency: Reliable across different prompts
- Production-Ready: Balanced quality-speed

Frontend Application



Key Challenges & Solutions

- Limited GPU Memory
  Freeze 98.9% params
- Large Dataset
  Domain filtering → 4K subset
- Hyperparameter Selection
  Systematic 9-config test + FID metric
- Training Stability
  Low LR, 2 epochs, gradient clipping

Any Questions?