Animal Posture Classification with Transfer Learning

Chen Qiao, Johann Verolet

April 5, 2024

Table of Contents

- Introduction: Background & Problem Definition
- Dataset Overview and Data Cleaning Procedures
- Model architectures: ResNet50 & ViT
- Training from Scratch vs. Transfer Learning
- Data Augmentation: Manifold Mixup
- Synthetic data: Stable Diffusion
- Results Comparison

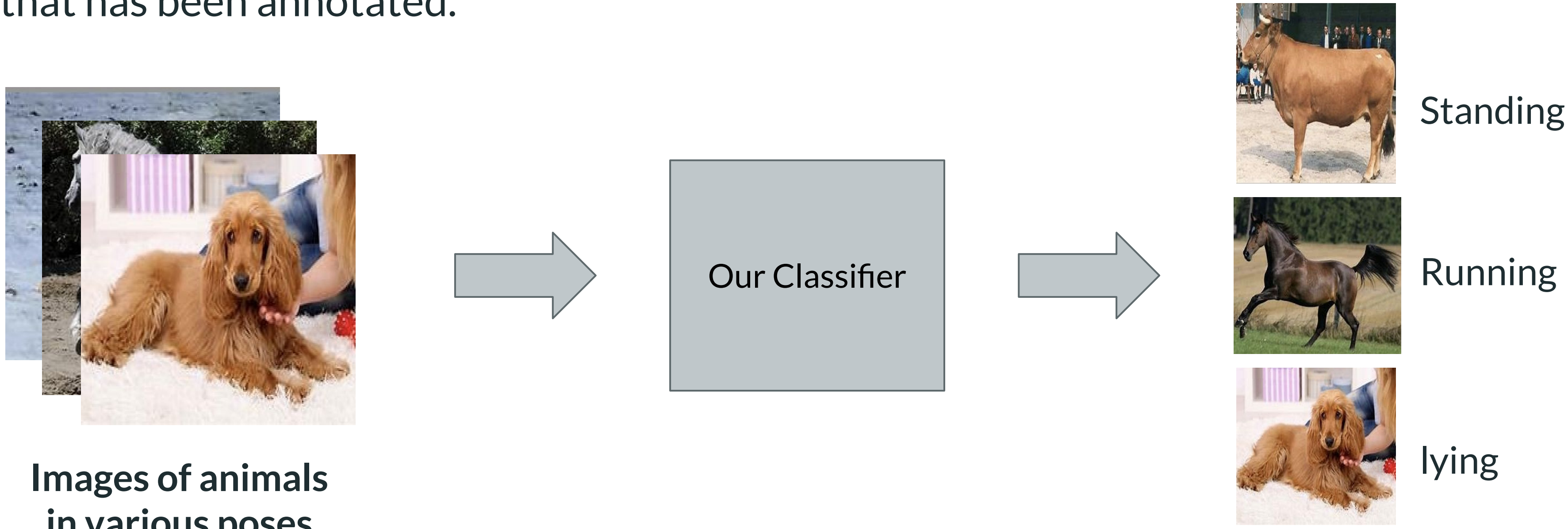
Background

Animal posture and behavior analysis are crucial not only in the field of wildlife preservation and agriculture but also in pet care and veterinary science, as it helps identify early signs of diseases in animals.

Our project aims to develop an Animal Posture Classification System. This system automatically classifies animals into different classes based on their posture such as lying, running, sitting, walking/standing.

Problem Definition

Achieving accurate posture classification in animals despite having a limited dataset that has been annotated.



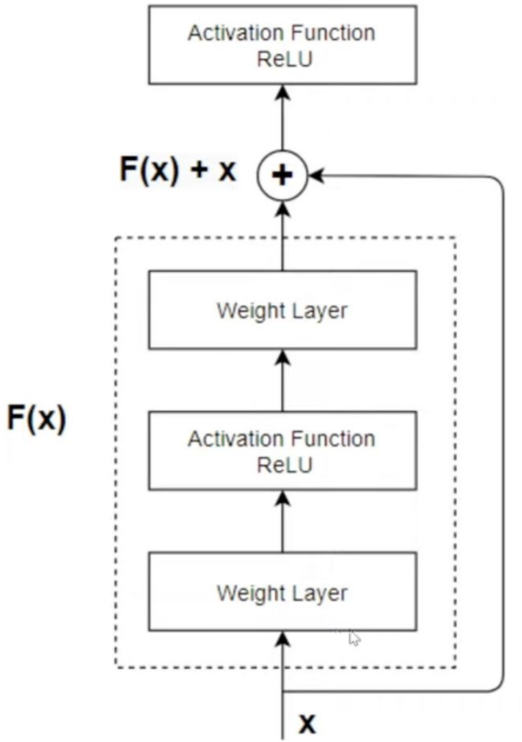
Images of animals in various poses

Results

Dataset

- We use the dataset Animals with Skeleton Key Points and Action Mark from Kaggle as our data source. It comprises **967** images of six different animal categories: cat, dog, horse, cow, sheep, and goat.
- Following a meticulous review and removal of unsuitable images, our animal posture dataset comprises **819** images, ensuring a high-quality foundation of our project.
- Dataset is unbalanced: 88 lying, 70 running, 67 sitting, 592 walking/standing

Network Architecture



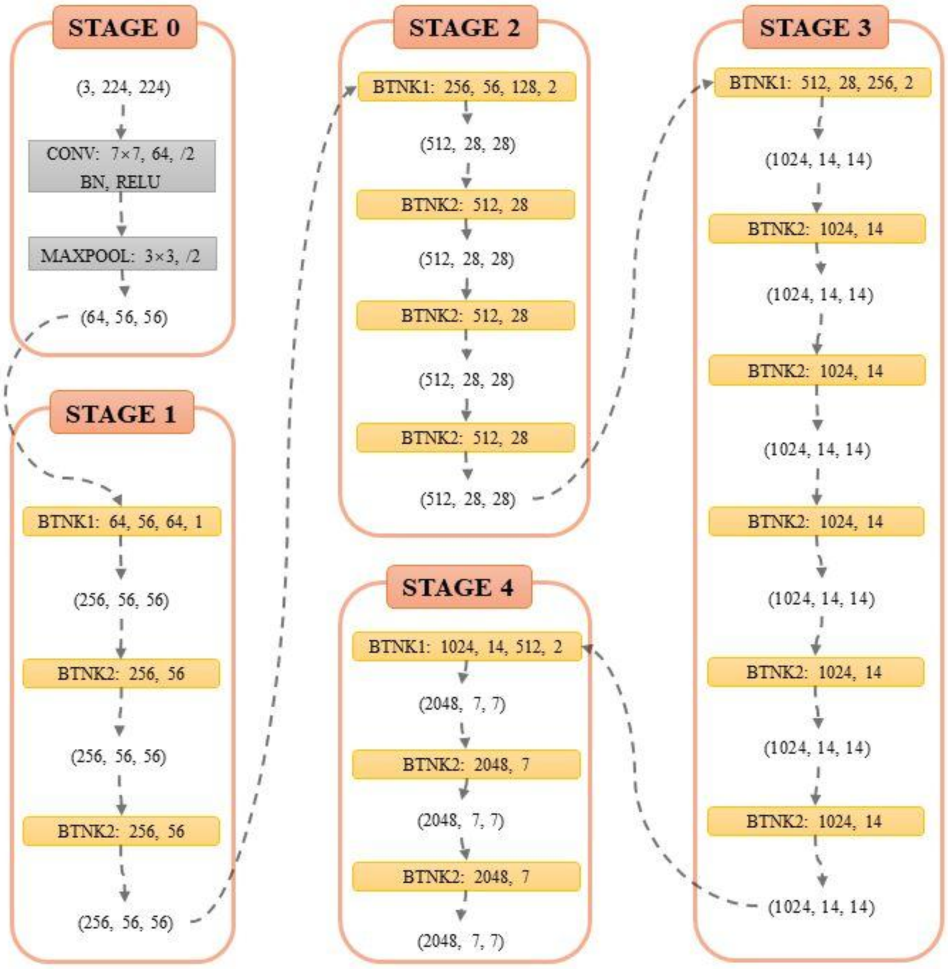
ResNet50 Residual Block

- First Weight Layer
- First Activation Function (ReLU)
- Second Weight Layer
- Skip Connection / Shortcut (F(x) + x)
- Second Activation Function (ReLU) after Addition

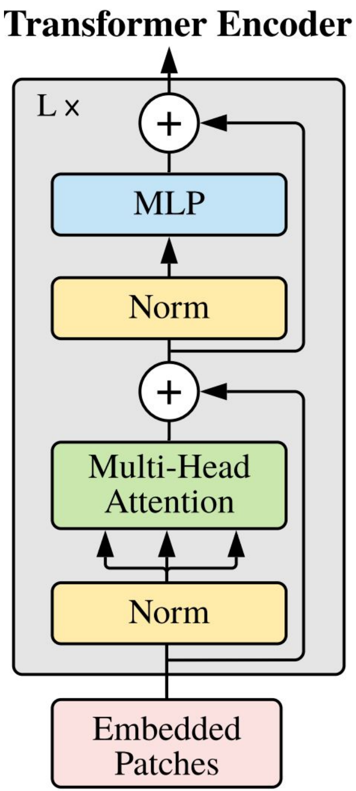
Network Architecture

ResNet50

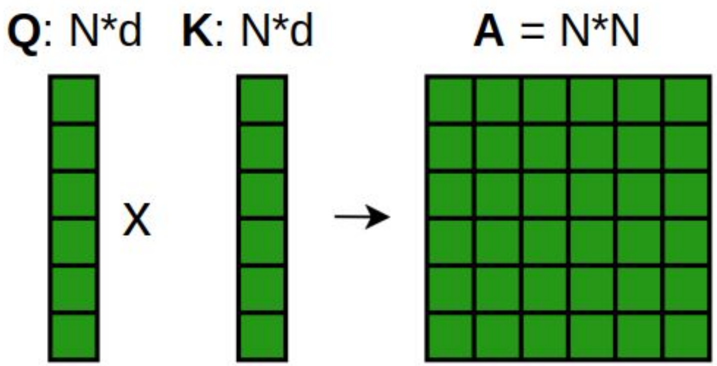
- Stage 0: extract low-level features
- Stage 1 - 4: extract and analyze features at varying levels



Recap: Transformer and Attention



Input sequence -> Q, K, V

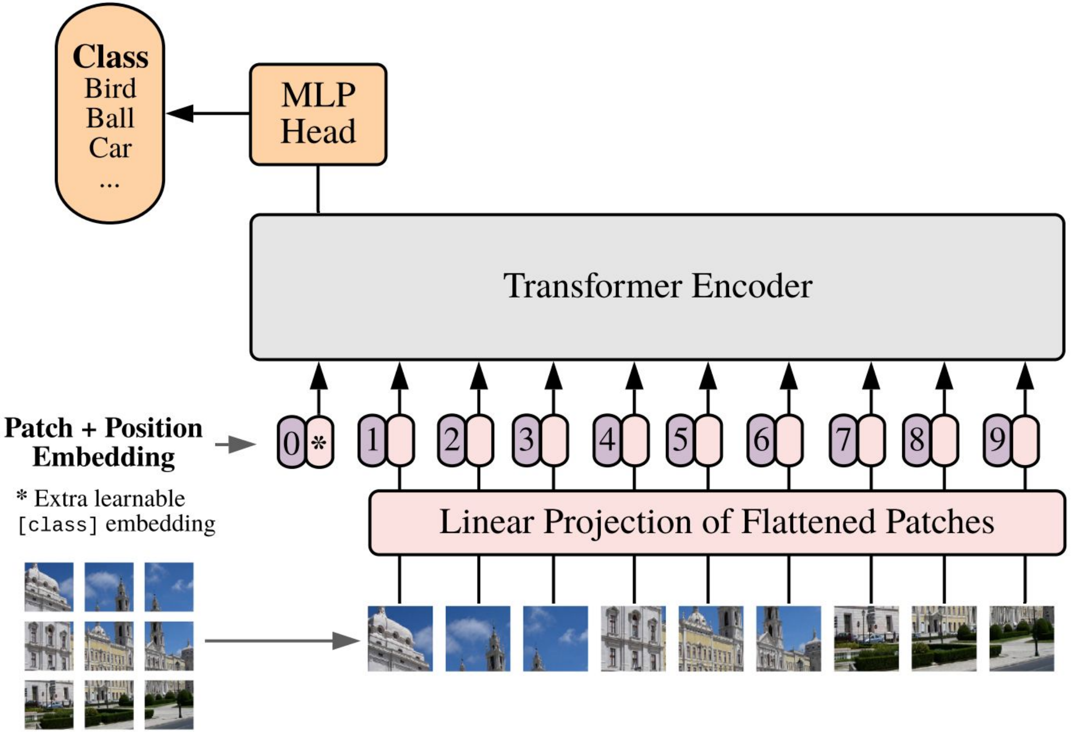


Attention: Pairwise inner product between each element in a sequence -> Attention matrix

Problems:

- Attention scales with **O(N^2)**
- **Images** are “long” sequences: ~ O(256^2)^2

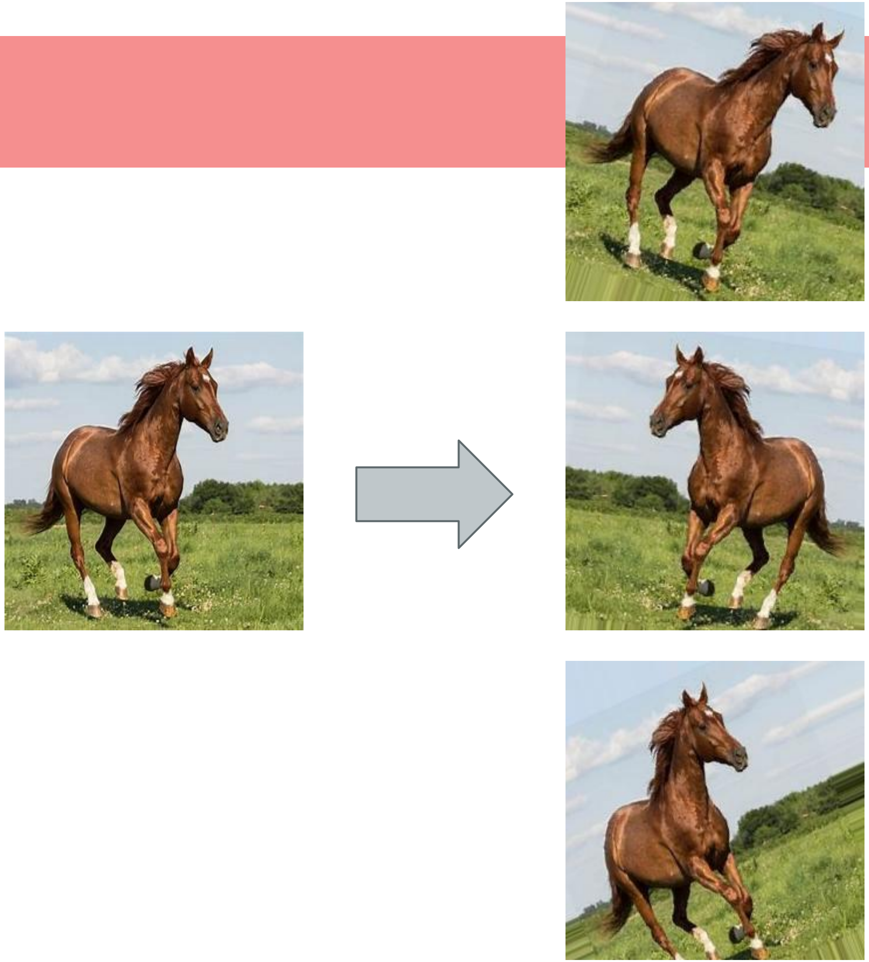
ViT: Vision Transformer



- **Not local** attention over **pixels**
- **Global** attention over **patches**
- Patches: 16x16
- **Patches** are unrolled/flattened
  - 16x16 = vector of length 256
  - Linear projection
- Pos. embeds are **learnable**
- Standard **Transformer**

Data Augmentation

- Random rotate
- Random shift
- Random shear transform
- Random zoom in
- Random horizontal flip



Baseline Results: Model from Scratch

ViT (10 epochs)

- Acc: 69.42% (no DA)
- Acc: 70.11% (DA)

Lie: 10%

Run: 0%

Sit: 13%

Walk/stand: 98%

ResNet (10 epochs)

- Acc: 70.11% (no DA)
- Acc: 70.11% (DA)

Lie: 0%

Run: 0%

Sit: 0%

Walk/stand: 100%

Transfer Learning

Pretrained Model Integration

- 2 types of experiments:
  - Pretrained layers set to **non-trainable** and **trainable**

Custom Layers for Classification

- Add classification head (linear layer)
- Output: 4 classes with sigmoid/softmax function

Both ResNet50 and ViT are pretrained on ImageNet.

Baseline results: Pretrained on ImageNet

ViT (10 epochs)

- Train pretrained layers
- Acc: **92.64%** (no DA) ↑22.53%
- Acc: 88.22% (DA)

Lie: 70% (↑60%)

Run: 75% (↑75%)

Sit: 75% (↑62%)

Walk/stand: 97% (↓1%)

ResNet (10 epochs)

- Freeze pretrained layers
- Acc: 85.06% (no DA)
- Acc: **86.20%** (DA) ↑16.09%

Lie: 80% (↑80%)

Run: 63% (↑63%)

Sit: 63% (↑63%)

Walk/stand: 93% (↓7%)

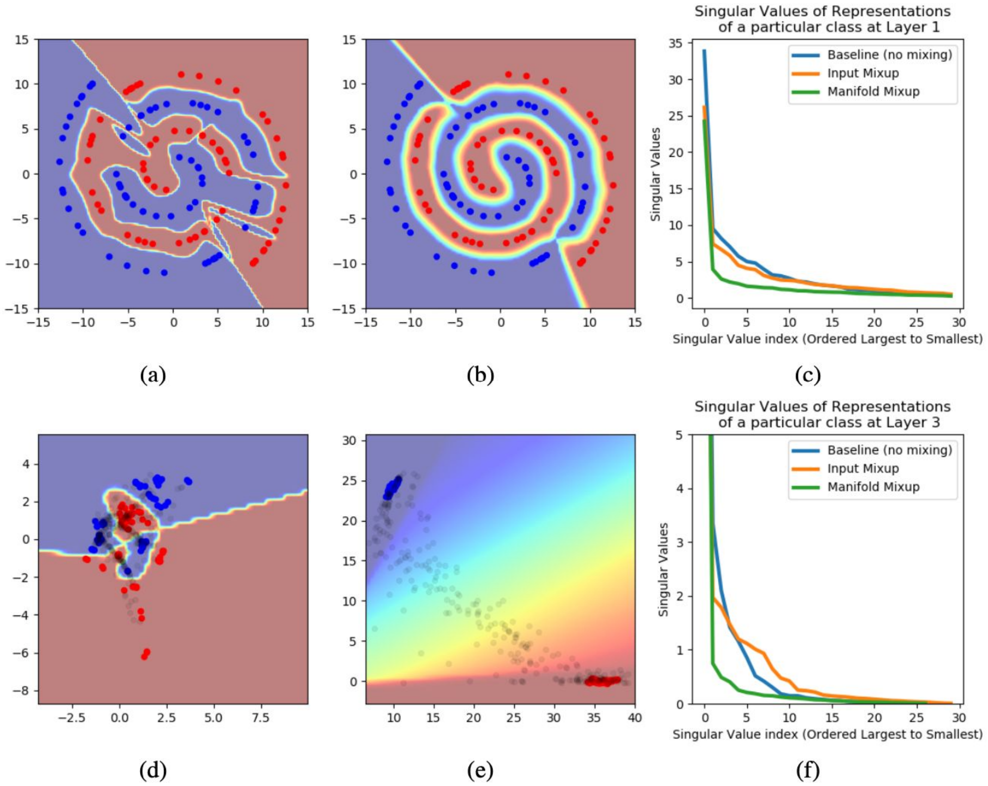
Manifold Mixup

Manifold Mixup Technique

- Select a layer k from the network
- Two random mini batches of data (x, y), (x', y') up to layer k, resulting in intermediate mini batches (g(x), y), and (g(x'), y'))
- Perform linear interpolation on these minibatches:

$(\tilde{g}, \tilde{y}) := (\mathrm{Mix}_\lambda(g(x), g(x')), \mathrm{Mix}_\lambda(y, y'))$ Here, $\mathrm{Mix}_\lambda(a, b) = \lambda \cdot a + (1 - \lambda) \cdot b,$

Manifold Mixup



- Smoothens decision boundaries
- Encourages broader region
- Flattens the representations

Results after Manifold Mixup

ViT (15 epochs)

- Acc: 94.49% (↑1.85%)

Lie: 80% (↑10%)

Run: 75% (-)

Sit: 88%(↑13%)

Walk/stand: 97% (-)

ResNet (20 epochs)

- Acc: 89.09% (↑2.89%)

Lie: 83% (↑3%)

Run: 50% (↓13%)

Sit: 71% (↑8%)

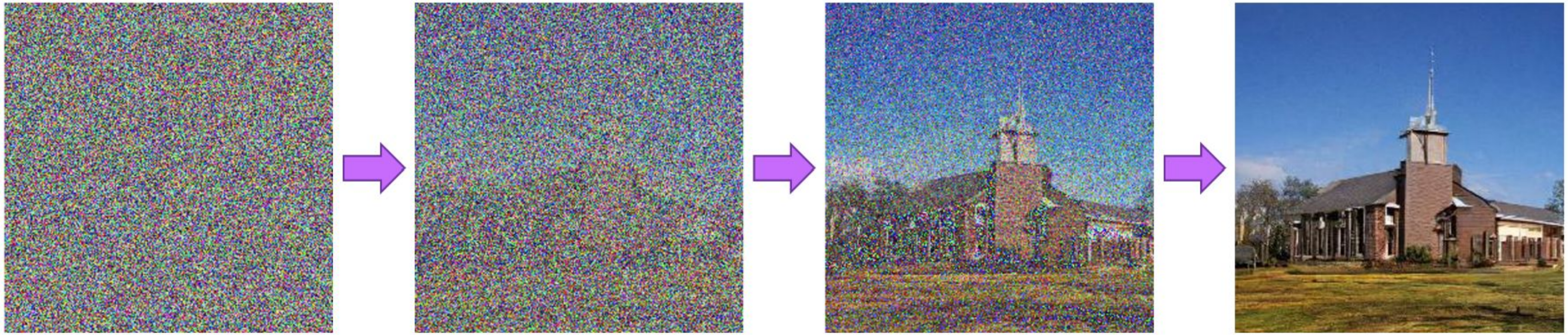
Walk/stand: 97% (↑4%)

Stable Diffusion

- Generate images conditioned on text prompts
- "a photograph of an astronaut riding a horse"

Diffusion models

- Denoise random gaussian noise step by step
- Operate in **pixel space** -> slow
- **Latent diffusion**:
    - Diffusion over lower dimensional **latent space**
    - Components: **VAE**, **U-Net**, text encoder (e.g. **CLIP**)



Diffusion models: Training

VAE

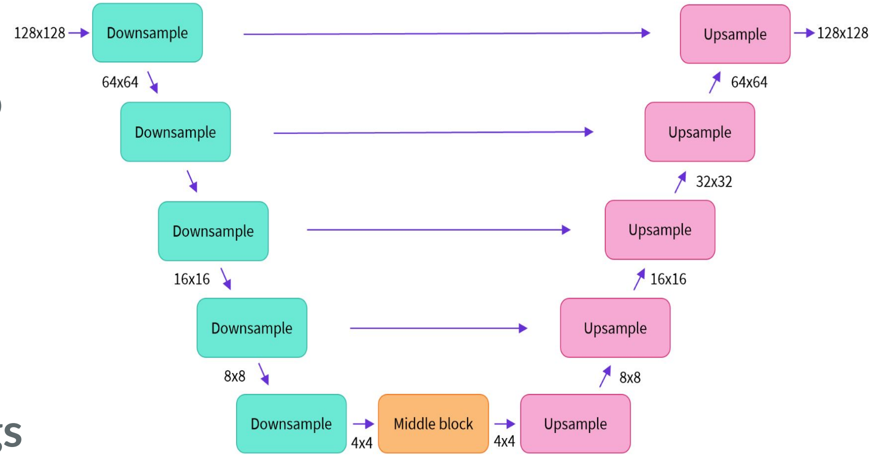
- Encoder: Convert image into latent representation -> U-Net input
- Decoder: Transform latent representation back into image

Text-encoder

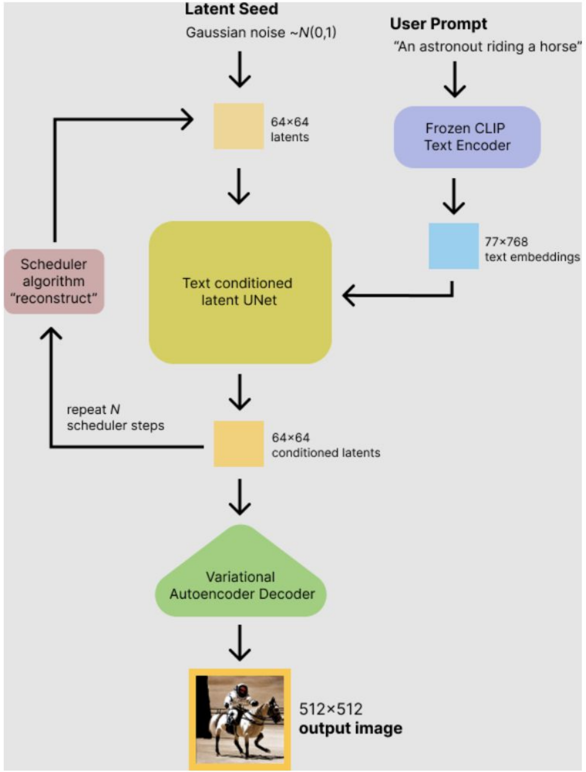
- Transforms prompt “running horse” into embedding

U-Net

- Output predicts **noise residuals**
  -> compute denoised image
- Output conditioned on **text-embeddings** (cross-attention)



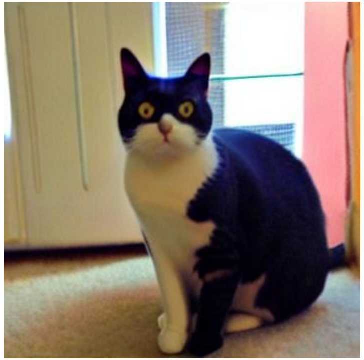
Diffusion models: Inference



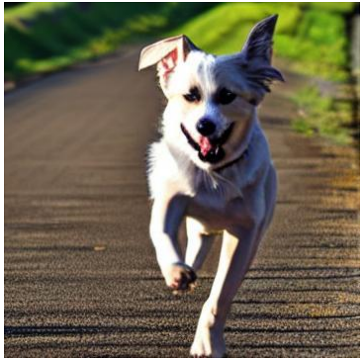
- **Input:** Gaussian noise & prompt (no decoder)
- U-Net **iteratively** denoises latent image representation (conditioned on prompt)
- **Output (noise residual):** compute denoised image representation
- Denoising process: ~50 steps
- VAE **Decoder** converts latent rep. Into image

Synthetic Data

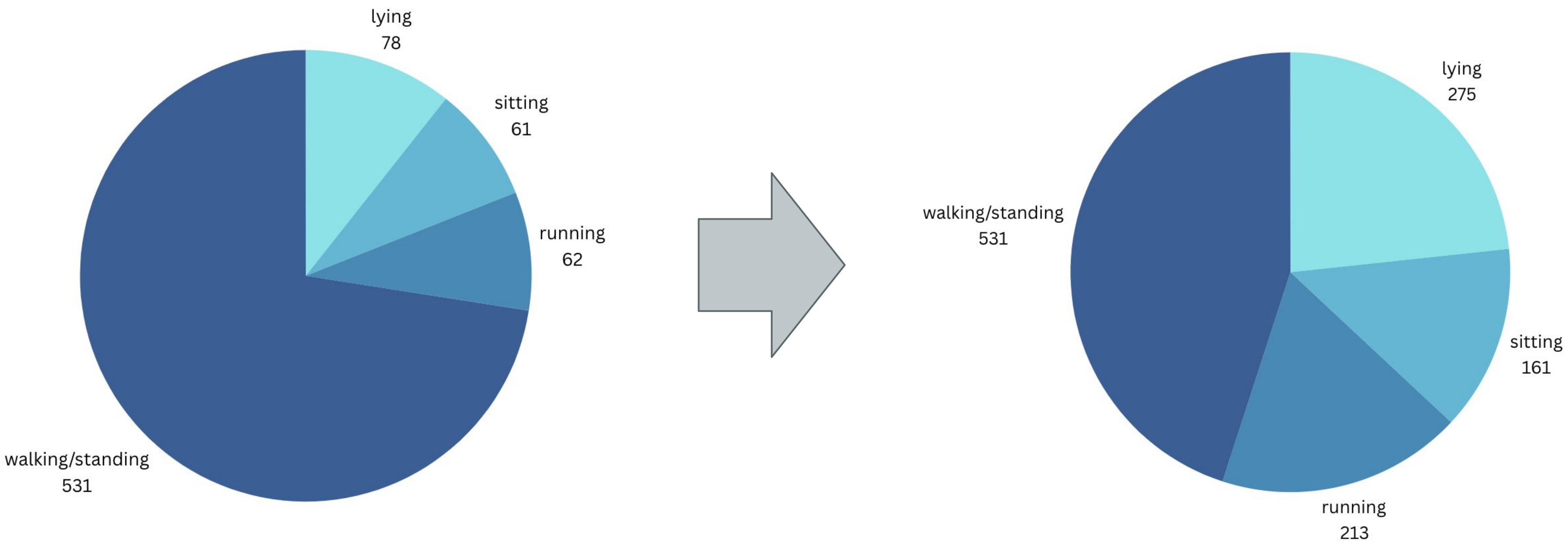






Integrating Synthetic Data: Before & After



Results after adding synthetic data

ViT (10 epochs)

- Acc: **94.63%** (↑1.99%)

Lie: 100% (↑30%)

Run: 75% (-)

Sit: 88% (↑13%)

Walk/stand: 97% (-)

ResNet (20 epochs)

- Acc: 87.27% (↑1.07%)

Lie: 83.33% (↑3.33%)

Run: 71.43% (↑8.93%)

Sit: 78.57% (↑16.07%)

Walk/stand: 90.75% (↓2.96%)

Results Comparison

	Baseline Accuracy	Transfer Learning Accuracy	Manifold Mixup Accuracy	Synthetic Data Accuracy
ViT	70.11%	92.64% (fft)	94.49% (fft)	**94.63%** (fft)
ResNet50	70.11%	86.20% (lp)	**89.09%** (lp)	87.27% (lp)

Using manifold mixup and synthetic data at the same time did not provide higher accuracy.

Reference

Yu, H., Xu, Y., Zhang, J., Zhao, W., Guan, Z., & Tao, D. (2021). AP-10K: A Benchmark for Animal Pose Estimation in the Wild. arXiv:2108.12617v2.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. IEEE, 109(1), 43-76.

Kumar, A., Raghunathan, A., Jones, R., Ma, T., & Liang, P. (2022). Fine-tuning can distort pretrained features and underperform out-of-distribution. arXiv preprint arXiv:2202.10054.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. International Conference on Learning Representations (ICLR).

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Courville, A., Lopez-Paz, D., Bengio, Y. (2018). Manifold Mixup: Better Representations by Interpolating Hidden States. International Conference on Machine Learning.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2021). High-Resolution Image Synthesis with Latent Diffusion Models. Conference on Computer Vision and Pattern Recognition (CVPR).

Thank you !