Multi-Modal Data Retrieval

Deep Learning Divas

December 10, 2025

Derek DeBlieck, Sabrina Fowler, Grace Hecke, Abby Veiman

The Problem

- How do we search for images using text queries?
  - Or find relevant captions for a given image?
- Challenge: Images and text live in different spaces
  - Images: pixel intensities, visual features
  - Text: words, semantic meanings
- Need: A shared representation to bridge modalities

Why Does This Matter?

- Real-world applications:
  - Image search engines
  - Content-based retrieval systems
  - Accessibility tools for visually impaired users
- Traditional approach: Treat modalities separately
  - Limited cross-modal understanding
- Our opportunity: Modern deep learning enables shared representations
  - More accurate retrieval
  - Better generalization across domains

Our Solution: Modernizing Correspondence Autoencoders

- Original Corr-AE (Feng et al., 2014)
  - Used Restricted Boltzmann Machines for feature extraction
  - Shared latent space for image and text
- Our modernized approach:
  - Replace RBMs with pretrained models:
    - ResNet-50 for image features
    - BERT for text embeddings
  - Compare multiple autoencoder architectures
  - Evaluate different alignment loss functions
- Dataset: Flickr8k (8,000 images, 5 captions each)

Outline

1. Background & Related Work
2. Methodology
   - Feature extraction
   - Autoencoder architectures
   - Loss functions
3. Experimental Results
   - Performance metrics
   - Architecture comparison
4. Conclusions & Future Work

Background: Original Corr-AE → Modern Approach

- Original Corr-AE (Feng et al., 2014)
  - RBMs for feature extraction
  - Shared latent space approach
- Our modernization
  - ResNet-50 (pretrained on ImageNet) for images
  - BERT (pretrained language model) for text
  - Modern features capture richer semantics
- Dataset: Flickr8k
  - 8,000 images, 5 captions each
  - Split: 70% train / 15% val / 15% test

Architecture Overview

- Correspondence Autoencoder (Corr-AE) with self-reconstruction:
  - Image encoder $\rightarrow$ 512-dim latent $\rightarrow$ Image decoder
  - Text encoder $\rightarrow$ 512-dim latent $\rightarrow$ Text decoder
  - Shared latent space enforced by alignment loss
- Two alignment loss functions compared:
  1. **MSE Loss:** Direct distance minimization between embeddings
  2. **Contrastive Loss:** Positive pairs close, negative pairs far
- Key question: Which alignment loss works better?
- Feature extraction:
  - Image: 2048-dim (ResNet-50) $\rightarrow$ 512-dim latent
  - Text: 768-dim (BERT) $\rightarrow$ 512-dim latent

Contrastive Loss: The Winning Approach

- Self-reconstruction with shared latent space
  - Image encoder $\rightarrow$ Image decoder
  - Text encoder $\rightarrow$ Text decoder
  - Shared 512-dim latent space
- Contrastive alignment loss (InfoNCE)
  - Brings paired embeddings together
  - Pushes unpaired embeddings apart
  - Temperature parameter τ controls sharpness
- Why it outperforms MSE loss:
  - Better handles negative samples
  - More robust to outliers
- Careful hyperparameter selection is critical

Evaluation Setup

- Task: Image retrieval given text caption
- Metrics:
  - **Recall@K:** Top-K retrieval accuracy (K $\in \{1, 5, 10\}$)
  - **Median Rank:** Median position of correct match (lower is better)
- Training setup:
  - Optimizer: Adam (lr=5e-4, weight decay=1e-5)
  - Batch size: 256, Max epochs: 20
  - Z-score normalization (per-modality)

Quantitative Results

Model	Recall@10	Median Rank
Corr-AE (MSE)	24.9%	42
Corr-AE (Contrastive, Baseline)	27.7%	30
Corr-AE (Optimized Contrastive)	**50.4%**	**10**

- **Contrastive loss + tuning achieves best performance**
- Hyperparameter tuning proved critical:
  - Loss weight balancing (λ_{recon}, $\lambda_{\text{contrastive}}$)
  - Learning rate and temperature (τ) optimization
  - Implementing dropout
- Proper optimization reveals true model capacity

Qualitative Results: Example Retrieval

- Optimized Corr-AE with contrastive loss
- Task: Retrieve images given text captions from validation set
- Model shows strong semantic understanding:
  - Correctly matches captions to images
  - Top-5 retrieved images are semantically similar
  - Captures fine-grained details (objects, scenes, actions)
- Median rank of 10 indicates most correct images in top-10
- 50.4% of queries retrieve correct image in top-10

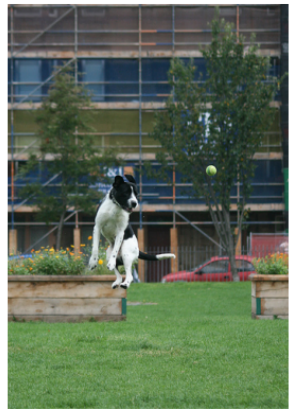
Qualitative Results: Example Retrieval

Caption: A black and white dog is attempting to catch a yellow and purple object in a low cut yard .

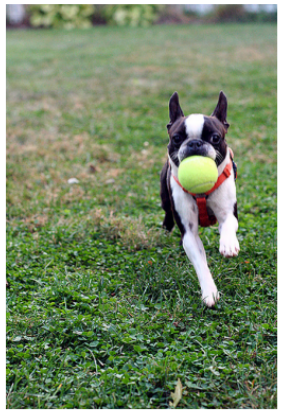
True Image

Rank 1 ✓

Rank 2



Rank 3



Rank 4

Rank 5

Key Finding: Loss Function + Hyperparameters Matter

- Contrastive loss outperforms MSE alignment
  - MSE: 24.9% Recall@10
  - Contrastive (baseline): 27.7% Recall@10
  - Contrastive (optimized): 50.4% Recall@10
- Critical hyperparameters for contrastive loss:
  - Loss weight balance: $\lambda_{\text{contrastive}}$ / λ_{recon}
  - Contrastive temperature τ
  - Dropout and learning rate
- Main takeaway: Proper loss function choice + tuning is critical
- Contrastive loss better captures semantic alignment

Why Does Contrastive Loss Win?

MSE Alignment Limitations

- Direct L2 distance minimization: $\mathcal{L}_{\text{MSE}} = ||z_{\text{img}} - z_{\text{txt}}||^2$
- Only considers positive pairs (matching image-caption)
- No explicit push against negative pairs
- Sensitive to outliers

Contrastive Loss Advantages

- Pulls positive pairs together, pushes negative pairs apart
- Temperature τ: Controls hardness of negative samples
- More robust optimization landscape
- Better semantic alignment in shared latent space

Key insight: Negative sample learning is critical for retrieval tasks

Hyperparameter Optimization Details

- Systematic grid search over:
  - Loss weights: $\lambda_{\text{recon}} \in \{0.5, 1.0, 2.0\}$
  - Contrastive weight: $\lambda_{\text{contrastive}} \in \{0.5, 1.0, 2.0\}$
  - Temperature: $\tau \in \{0.05, 0.07, 0.10\}$
  - Learning rate: {5e-4, 1e-3, 2e-3}
- Validation set used for selection
- Best configuration: $\lambda_{\text{recon}} = 1.0$, $\lambda_{\text{contrastive}} = 1.0$, $\tau = 0.07$, dropout $= 0.25$, batch size $= 256$, lr $= 5 \times 10^{-4}$
- Demonstrates importance of tuning for correspondence learning

Future Work

- Bidirectional retrieval
  - Image $\rightarrow$ text (5 captions per image)
- Scale to larger datasets
  - Flickr30k, MS COCO
- Compare with state-of-the-art
  - CLIP, ALIGN
- Architectural improvements
  - Attention mechanisms
  - Transformer-based encoders
- Apply to other domains
  - Medical imaging, audio-text retrieval

Conclusions

- Successfully modernized Corr-AE framework
  - Replaced RBMs with ResNet-50 and BERT
- Key result: Optimized Corr-AE achieves **50.4% Recall@10**
  - Median rank of 10
- Hyperparameter tuning is critical
  - Proper optimization reveals true model capacity
- Demonstrates viability of modernized correspondence learning
- Framework applicable to other multimodal retrieval tasks

Thank you! Questions?

Backup: Training Details

- Optimizer: Adam
  - Learning rate: 5e-4
  - Weight decay: 1e-5
- Batch size: 256
- Max epochs: 20 (with early stopping)
- Normalization: Z-score (per-modality)
  - Computed on training set
  - Applied to all splits
- Early stopping on validation loss

Backup: Architecture Specifications

- Image encoder: $2048 \rightarrow 1024 \rightarrow 512$
- Text encoder: $768 \rightarrow 512 \rightarrow 512$
- Decoders: Symmetric (reverse of encoders)
- Activations: ReLU between layers
- Loss components:
  - Reconstruction: MSE
  - Alignment: Contrastive (InfoNCE)
  - Optimized weights: $\lambda_{\text{recon}} = 1.0$, $\lambda_{\text{contrastive}} = 1.0$
  - Temperature (τ): 0.07

Backup: Full Results Table

Model	R@1	R@5	R@10	MedR
Corr-AE (MSE)	5.2%	16.7%	24.9%	42
Corr-AE (Contrastive, Baseline)	5.0%	17.4%	27.7%	30
Corr-AE (Optimized Contrastive)	13.6%	37.2%	50.4%	10

All metrics on test set (1,214 images, 6,070 captions).

Backup: Hyperparameter Search Space

- Grid search parameters:
  - Reconstruction loss weight: $\lambda_{\text{recon}} \in \{0.5, 1.0, 2.0\}$
  - Contrastive loss weight: $\lambda_{\text{contrastive}} \in \{0.5, 1.0, 2.0\}$
  - Temperature: $\tau \in \{0.05, 0.07, 0.10, 0.15\}$
  - Learning rate: {5e-4, 1e-3, 2e-3}
- Total configurations explored: 11
- Selection criterion: Best validation Recall@10
- Computational cost: 40-60 minutes total on M4 Mac CPU (2-3 sec/epoch $\times$ 20 epochs $\times$ 11 configs)