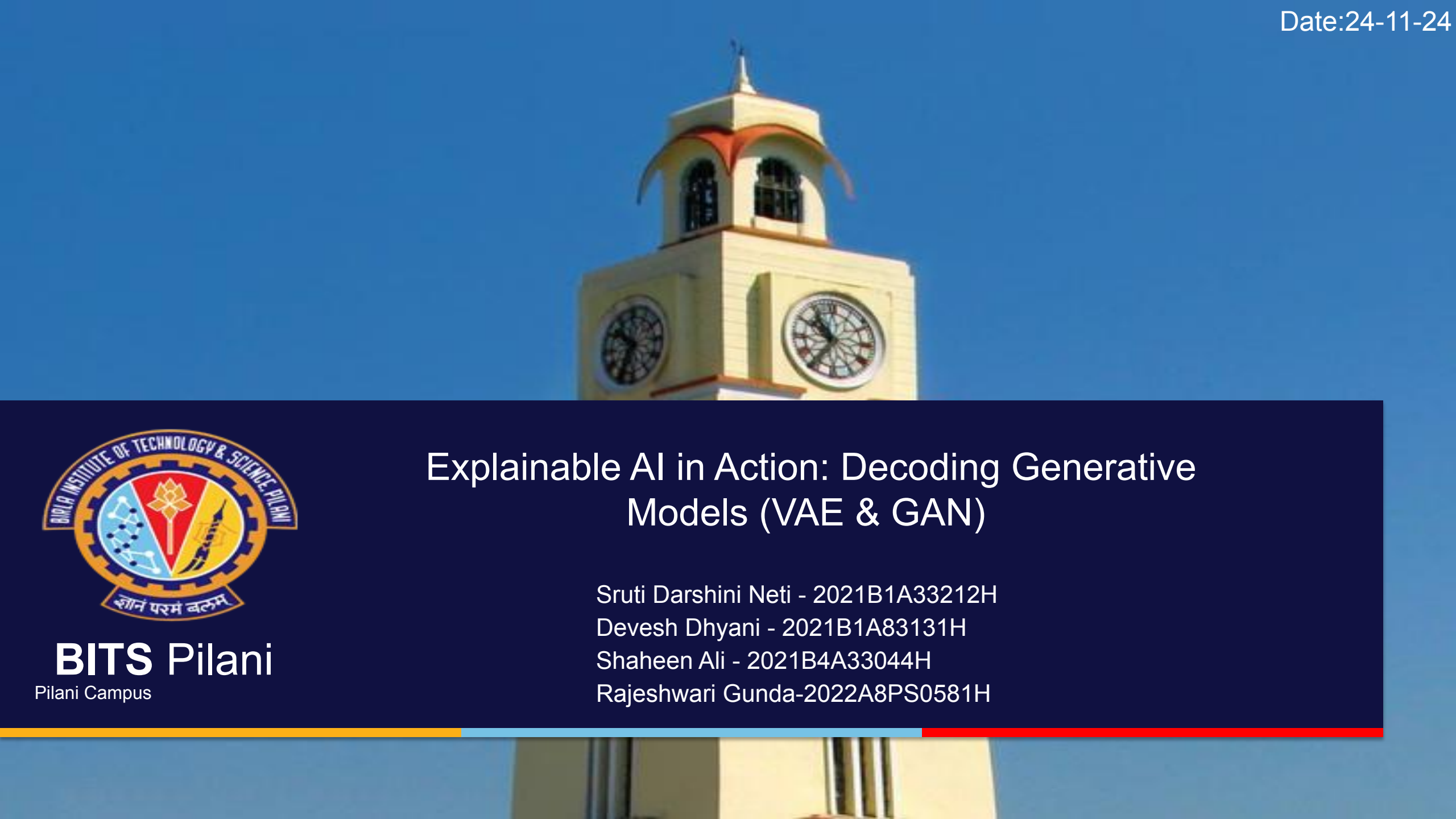
Date:24-11-24



Explainable AI in Action: Decoding Generative Models (VAE & GAN)

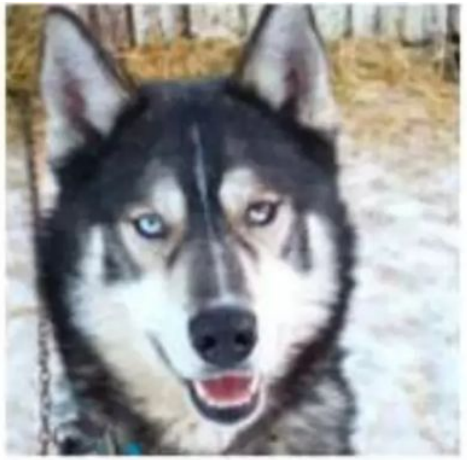
Sruti Darshini Neti - 2021B1A33212H
Devesh Dhyani - 2021B1A83131H
Shaheen Ali - 2021B4A33044H
Rajeshwari Gunda-2022A8PS0581H

BITS Pilani
Pilani Campus

Introduction

- Interpretability in Machine Learning can be seen as :
  “Trust that the model is predicting a certain value for the "right reasons".”
- Interpretability is key to ensure the social acceptance of Machine Learning algorithms in our everyday life.
- **Would you trust a model that’s accurate but unexplainable?**

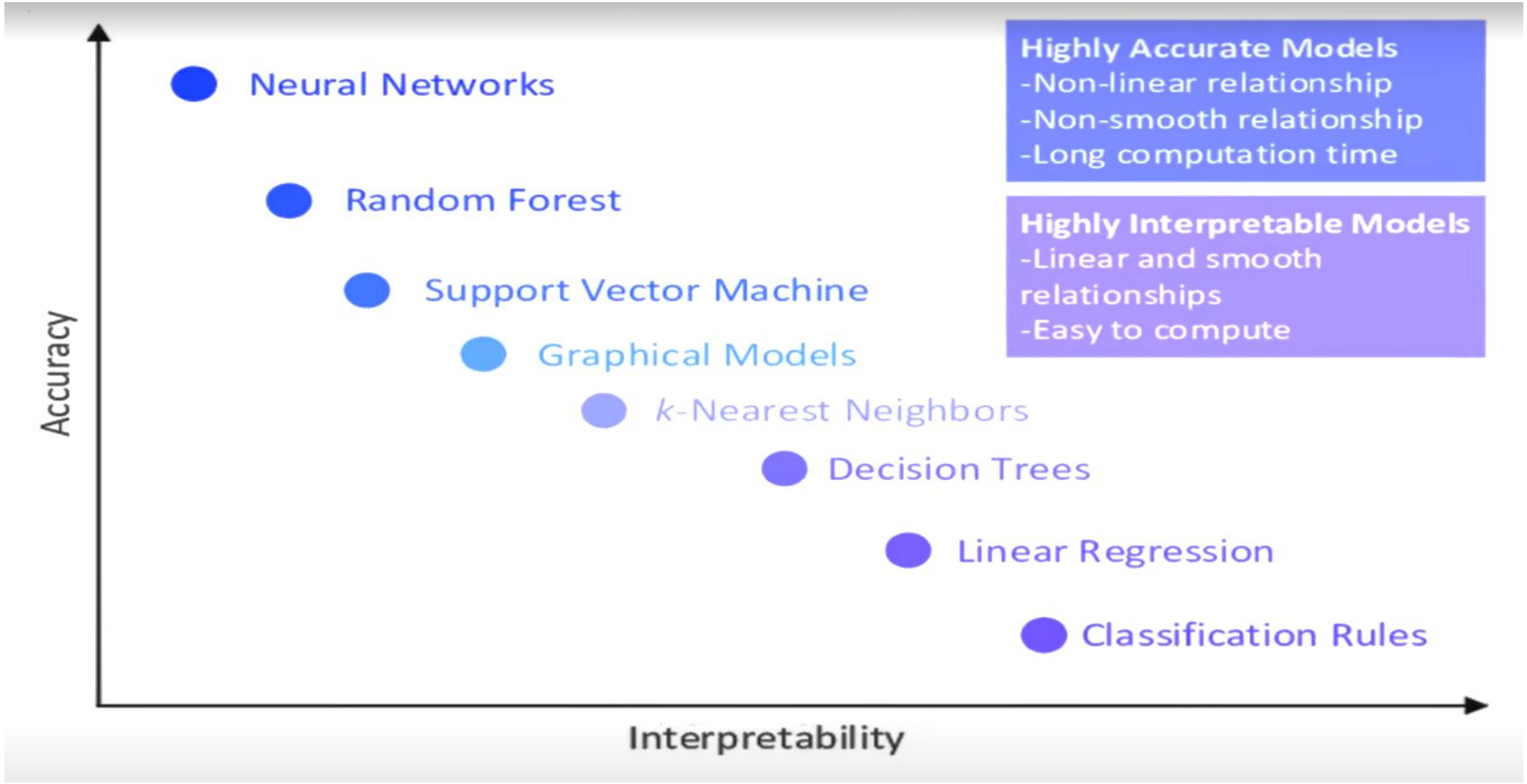
  In the Husky vs. Wolves experiment, researchers found their high-accuracy model relied on snowy backgrounds in wolf images, not the animals themselves.



(a) Husky classified as wolf

(b) Explanation

WHY NEURAL NETWORKS?



Course Objectives

- **Dimensionality Reduction**: Simplify latent spaces for visualization and analysis using PCA, t-SNE, and UMAP.
- **Neuron Activation Analysis**: Interpret model decisions with GradCAM, SHAP, LIME, Saliency Maps, and LRP.

1. **Dimensionality Reduction on Latent Spaces**
   - **PCA** (Principal Component Analysis)
   - **t-SNE** (t-Distributed Stochastic Neighbor Embedding)
   - **UMAP** (Uniform Manifold Approximation and Projection)
2. **Neuron Activations**
3. **GradCAM** (Gradient-weighted Class Activation Mapping)
4. **SHAP** **(**SHapley Additive exPlanations)
5. **LIME** (Local Interpretable Model-Agnostic Explanations):
6. **Saliency Maps**
7. **LRP** (Layer-wise Relevance Propagation)

Mid Semester Work Summary

Dimensionality Reduction on Latent Spaces:

- **PCA**: Identified dominant latent directions and explained variance in VAE and GAN.
- **t-SNE**: Visualized latent clusters, highlighting structure in latent representations.
- **UMAP**: Enhanced interpretability, preserving local and global structure.

Neuron Activations:

- Explored layer-wise activations

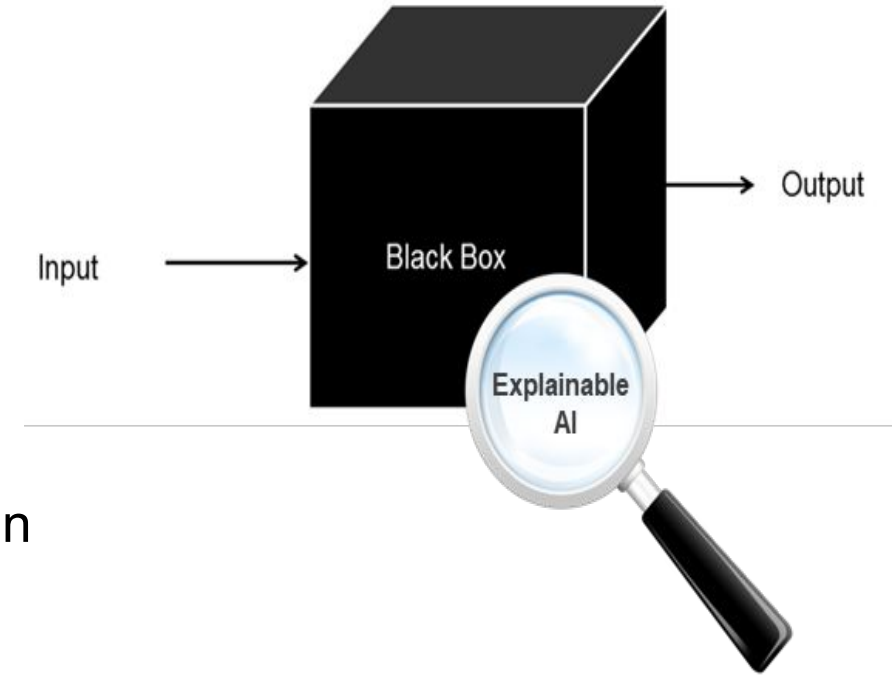
GradCAM:

- Visualized key image regions influencing latent representations in VAE and GAN.

Model Agnostic Methods

- **Universal Applicability**: Can be applied to any ML model (neural networks, random forests, etc.)
- **Post-hoc Analysis**: Applied after model training
- **Non-invasive**: Don't require modifying the original model
- **Input-Output Focused**: Analyze relationships between features and predictions

Examples : SHAP, LIME, LPR …



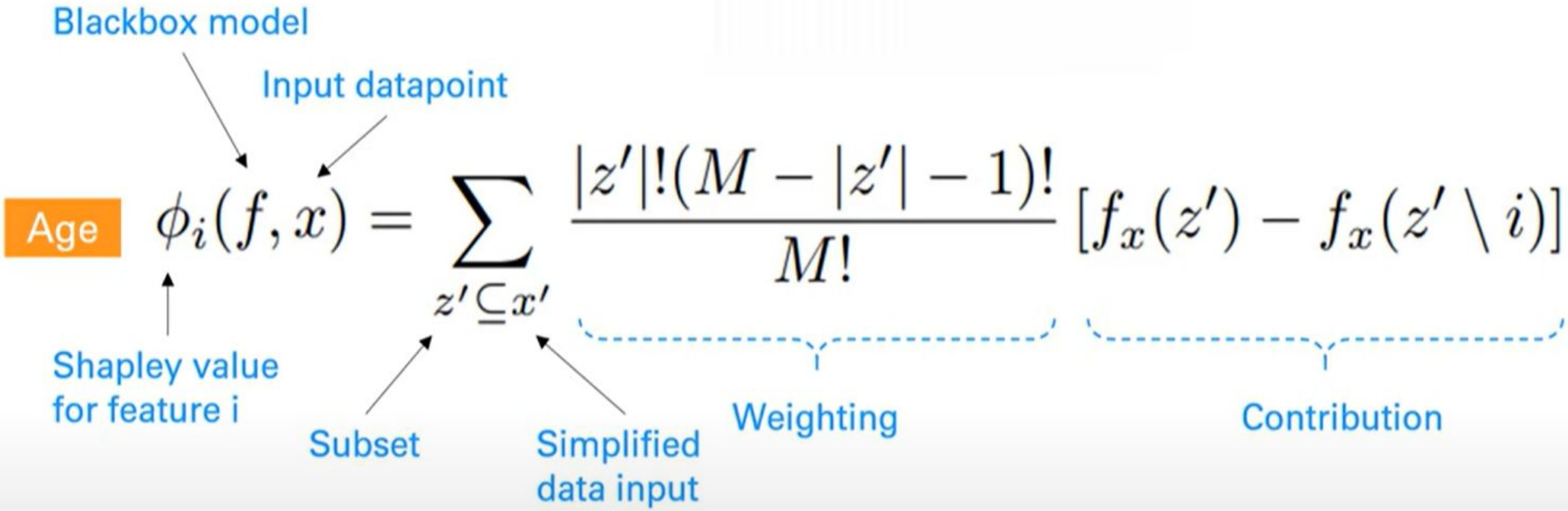
SHAP

- SHAP stands for **Shapley Additive Explanations**. It's a model-agnostic, efficient algorithm, to compute features contribution to a model output.
- With nonlinear black box models SHAP provides accurate and consistent features importance values
- SHAP works with any machine learning model, whether it's a tree-based model, neural network, or others.
- It allows meaningful, local explanations of individual predictions.
- SHAP borrows concepts from cooperative game theory: **The Shapley Values**

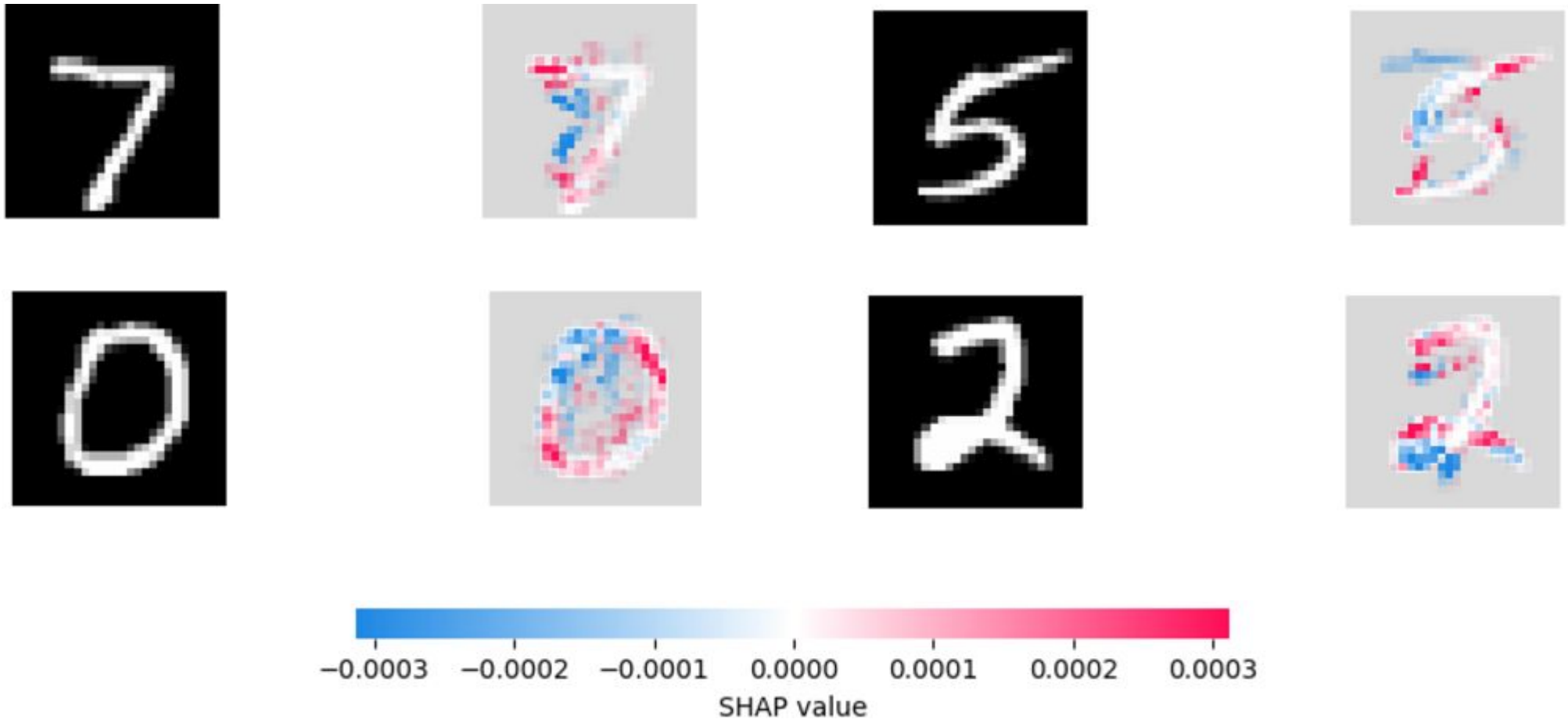
Scott Lundberg

Su-In Lee

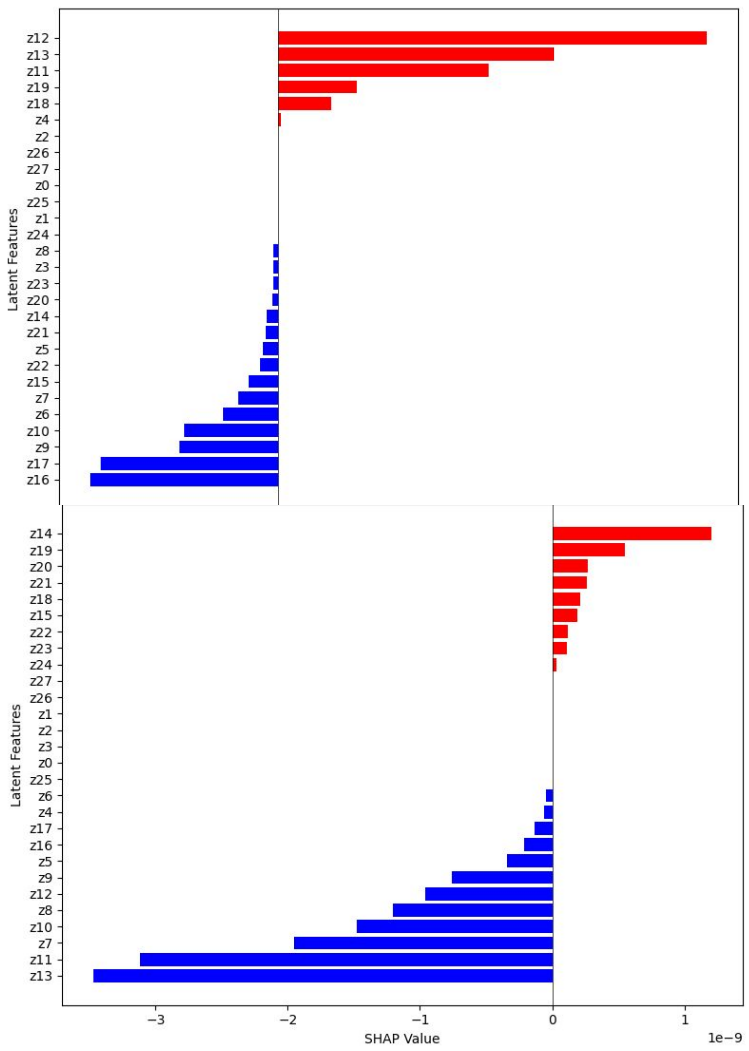
Math background of SHAP

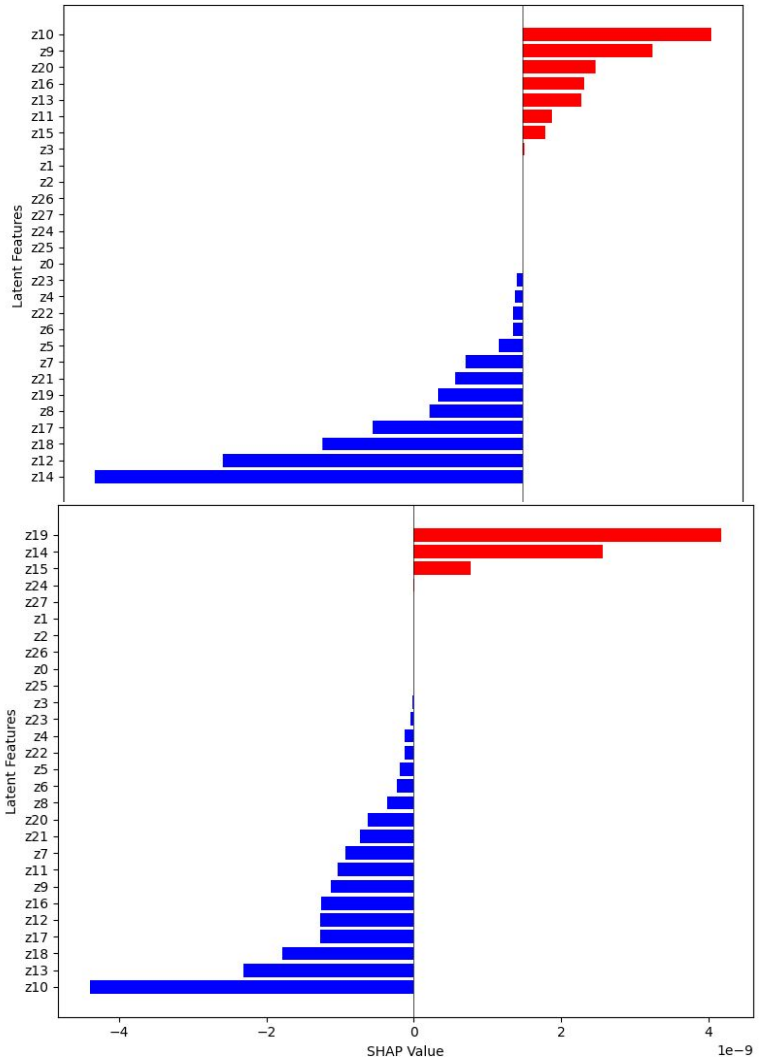


Results



Results


Latent Features
z12
z13
z11
z19
z18
z4
z2
z26
z27
z0
z25
z1
z24
z8
z3
z23
z20
z14
z21
z5
z22
z15
z7
z6
z10
z9
z17
z16
Latent Features
z14
z19
z20
z21
z18
z15
z22
z23
z24
z27
z26
z1
z2
z3
z0
z25
z6
z4
z17
z16
z5
z9
z12
z8
z10
z7
z11
z13
−3
−2
−1
0
1
SHAP Value
1e−9


Latent Features
z10
z9
z20
z16
z13
z11
z15
z3
z1
z2
z26
z27
z24
z25
z0
z23
z4
z22
z6
z5
z7
z21
z19
z8
z17
z18
z12
z14
Latent Features
z19
z14
z15
z24
z27
z1
z2
z26
z0
z25
z3
z23
z4
z22
z5
z6
z8
z20
z21
z7
z11
z9
z16
z12
z17
z18
z13
z10
−4
−2
0
2
4
SHAP Value
1e−9

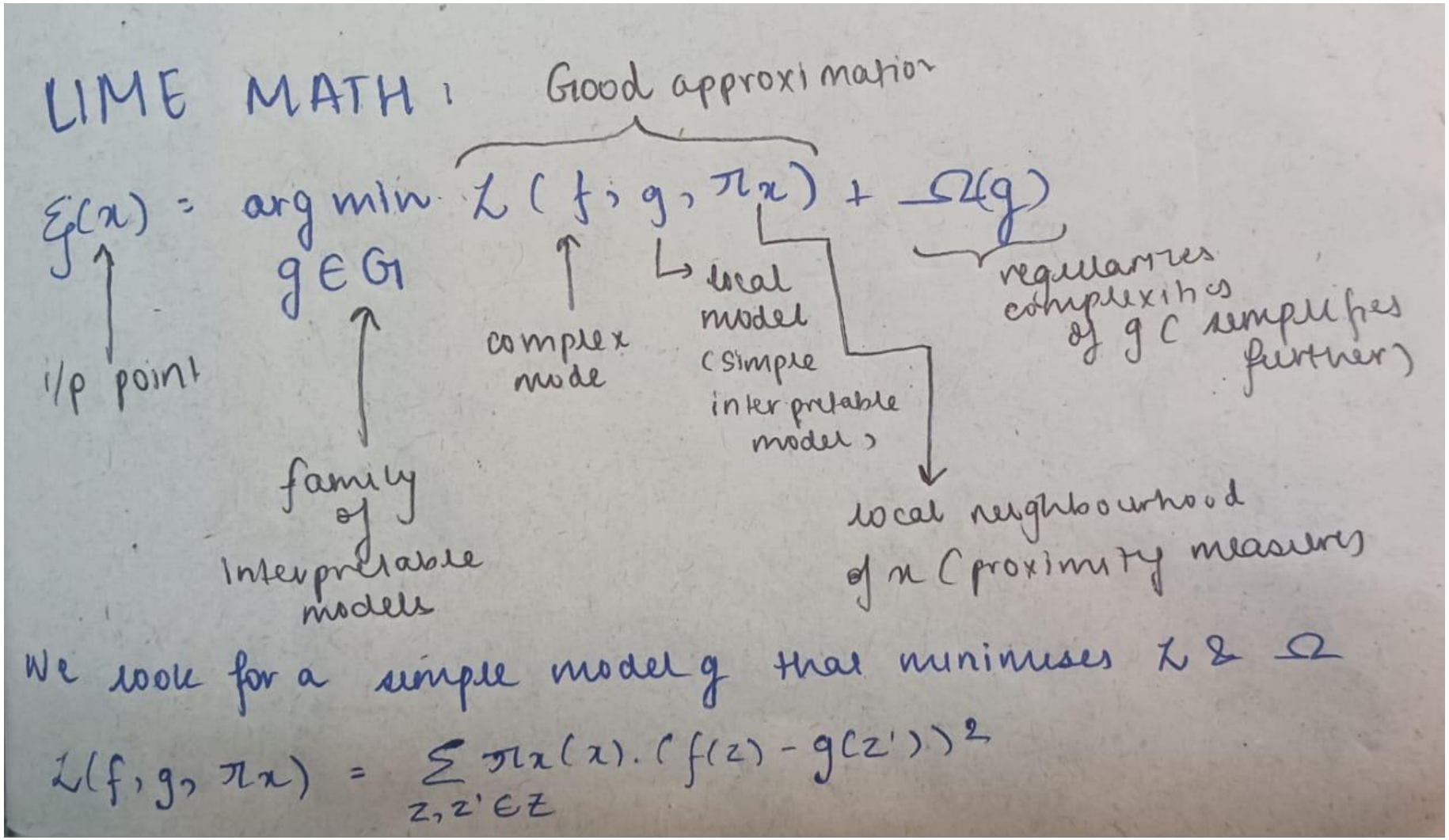
Locally Interpretable Model-Agnostic Explanations (LIME)

LIME explains individual predictions by approximating the complex model locally with a simpler, interpretable model. Instead of providing a global understanding of the model on the entire dataset, LIME focuses on explaining the model's prediction for individual instances.

Process:

- **Local Perturbation**: Generate variations of the input data point by slightly modifying feature values
- **Weighted Sampling**: Assign higher weights to perturbed samples closer to the original instance
- **Local Approximation**: Fit a simple model (usually linear) to predict the complex model's behavior in the local region
- **Feature Importance**: Extract feature contributions from the simple model's coefficients

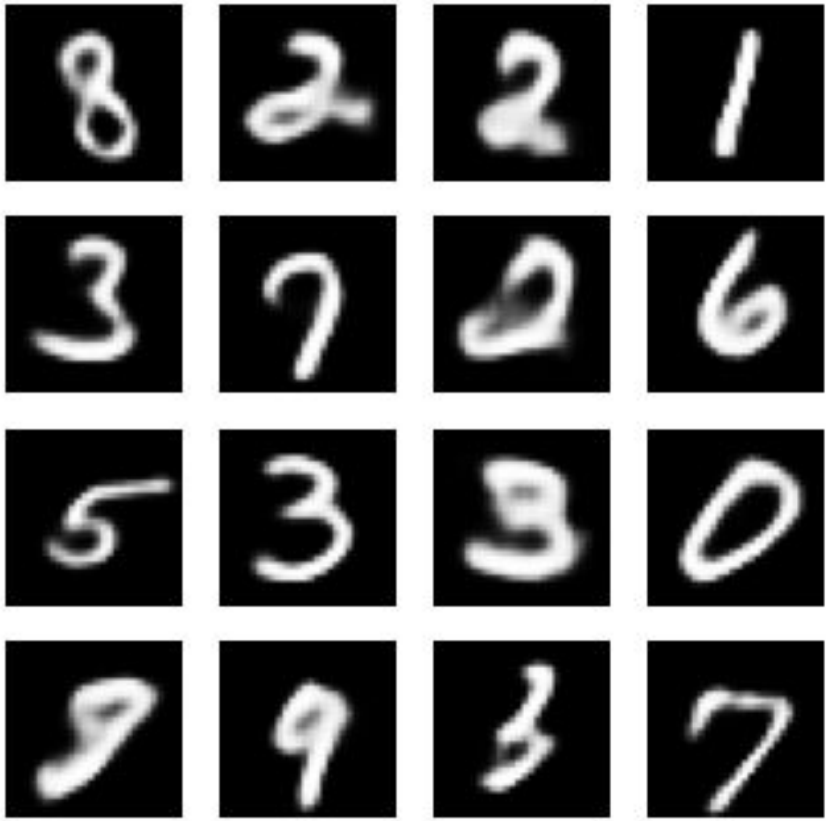
Mathematics behind LIME

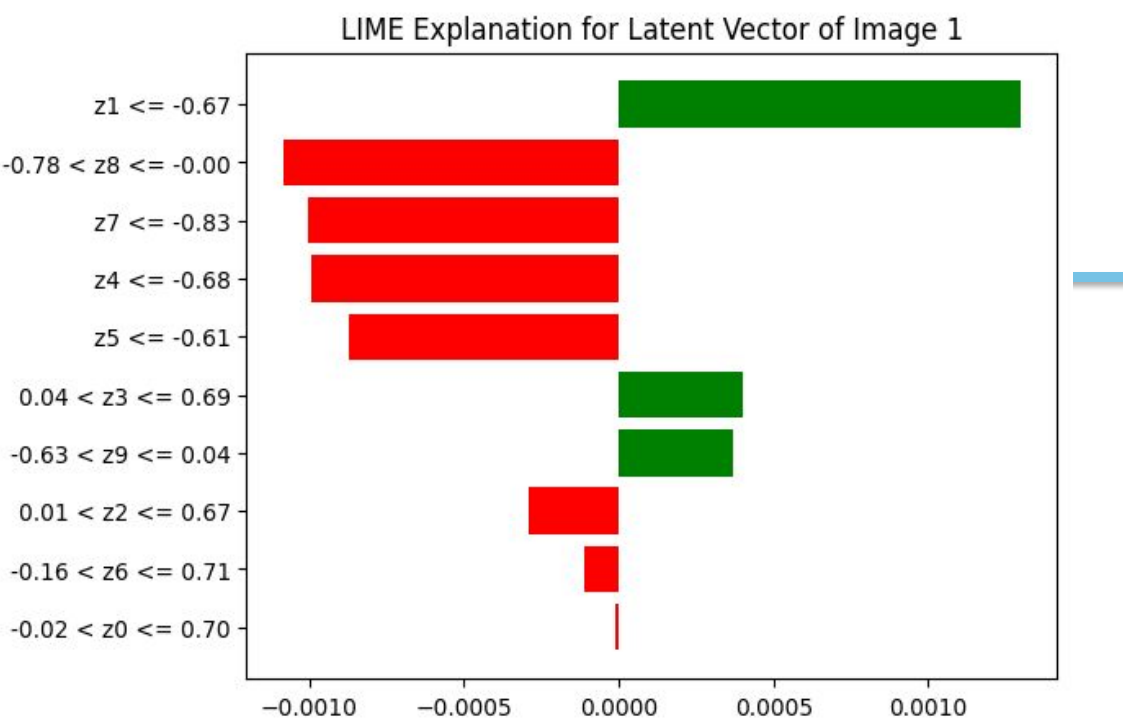


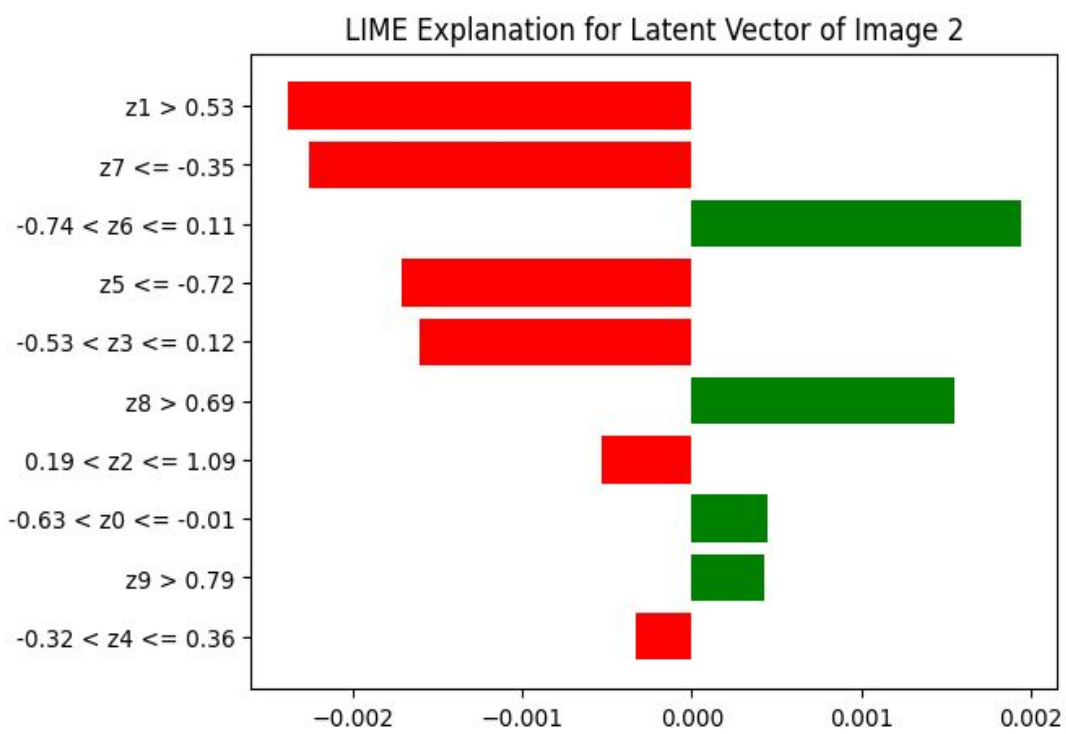
Latent Space - Output Mapping on VAE using LIME

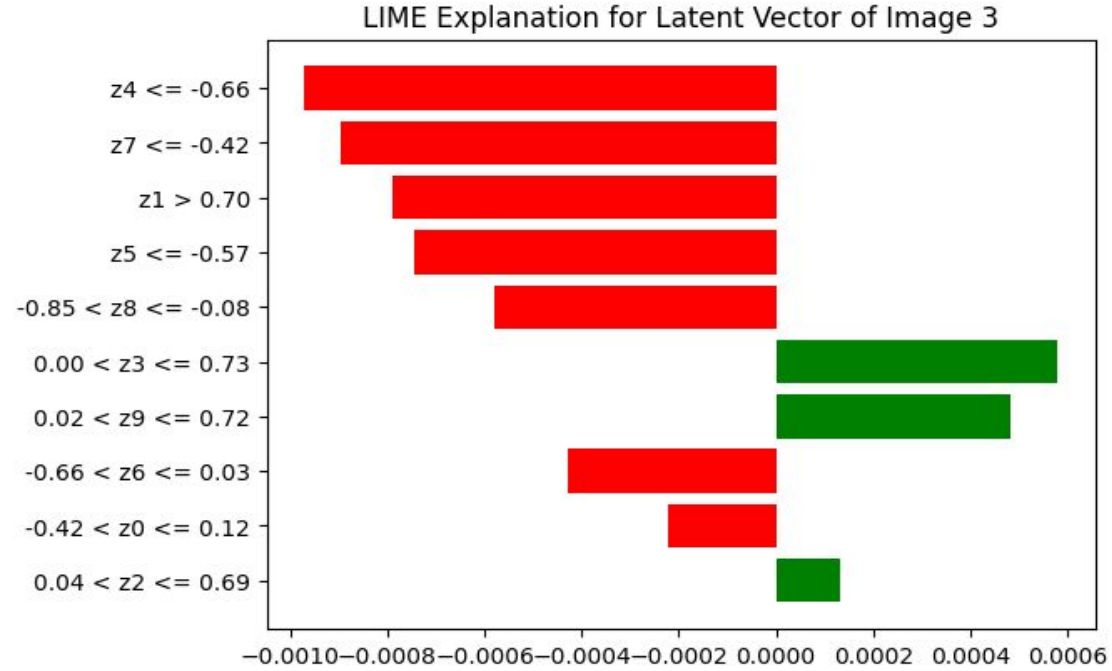
- In a VAE, each image is encoded into a latent vector (in this case, 10-dimensional, z0 through z9). Each dimension in this latent space captures some learned feature of the data.
- The inequalities (like "z3 > 0.81" or "z7 <= -0.82") represent conditions about these latent dimensions that LIME has identified as important for this specific input's representation.
- The values on the x-axis (like -0.0005, 0, 0.0005, etc.) represent the importance weights that LIME has assigned to each condition. Larger positive values (green bars) mean that satisfying that condition strongly supports the model's decision, while negative values (red bars) mean that satisfying that condition works against the model's decision.

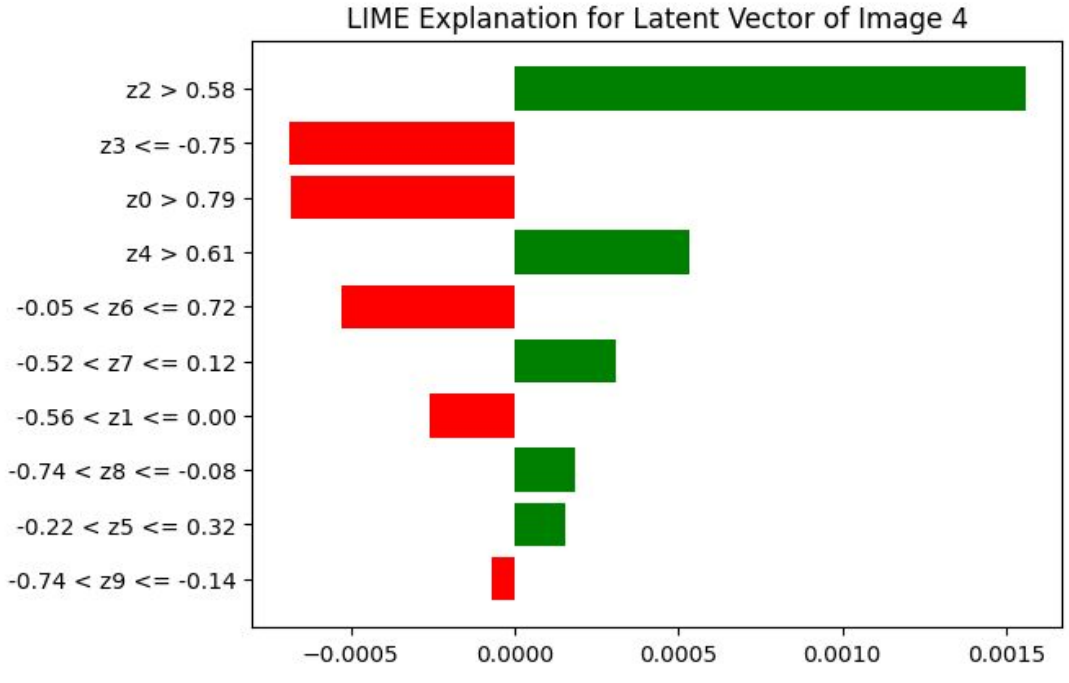
VAE output




LIME Explanation for Latent Vector of Image 1
z1 <= -0.67
-0.78 < z8 <= -0.00
z7 <= -0.83
z4 <= -0.68
z5 <= -0.61
0.04 < z3 <= 0.69
-0.63 < z9 <= 0.04
0.01 < z2 <= 0.67
-0.16 < z6 <= 0.71
-0.02 < z0 <= 0.70
−0.0010
−0.0005
0.0000
0.0005
0.0010


LIME Explanation for Latent Vector of Image 2
z1 > 0.53
z7 <= -0.35
-0.74 < z6 <= 0.11
z5 <= -0.72
-0.53 < z3 <= 0.12
z8 > 0.69
0.19 < z2 <= 1.09
-0.63 < z0 <= -0.01
z9 > 0.79
-0.32 < z4 <= 0.36
−0.002
−0.001
0.000
0.001
0.002


LIME Explanation for Latent Vector of Image 3
z4 <= -0.66
z7 <= -0.42
z1 > 0.70
z5 <= -0.57
-0.85 < z8 <= -0.08
0.00 < z3 <= 0.73
0.02 < z9 <= 0.72
-0.66 < z6 <= 0.03
-0.42 < z0 <= 0.12
0.04 < z2 <= 0.69
−0.0010
−0.0008
−0.0006
−0.0004
−0.0002
0.0000
0.0002
0.0004
0.0006


LIME Explanation for Latent Vector of Image 4
z2 > 0.58
z3 <= -0.75
z0 > 0.79
z4 > 0.61
-0.05 < z6 <= 0.72
-0.52 < z7 <= 0.12
-0.56 < z1 <= 0.00
-0.74 < z8 <= -0.08
-0.22 < z5 <= 0.32
-0.74 < z9 <= -0.14
−0.0005
0.0000
0.0005
0.0010
0.0015

Aspect	SHAP	LIME
Scope	Explains feature importance for both latent space and input-to-output mappings comprehensively.	Focuses on localized explanations, particularly effective for latent-to-output mappings.
Complexity	Handles complex feature interactions in the latent space well.	Simplifies explanations by approximating locally linear regions in latent space
Performance	Computationally expensive, especially when applied to high-dimensional latent spaces.	Faster and more computationally efficient for latent vector explanations.
Interpretability	Provides global interpretability for the entire latent space or model.	Offers intuitive, instance-specific explanations for individual latent vectors.
Usefulness in VAE	Highlights overall significance of features in generating reconstructions.	Maps specific latent variables to their contribution to reconstructions.
Limitations	May be slow for large-scale latent vectors or complex datasets like CelebA.	Depends on segmentation or sampling quality; results may vary across runs.

Model-Specific Methods

Model-specific explanation methods are interpretability techniques designed to work with particular types of machine learning models, most commonly deep neural networks. These methods leverage the internal structure, gradients, and architectural properties of the models to generate explanations for their predictions.

Key Characteristics:

1. Internal Access Requirements

- Direct access to model architecture
- Access to gradients and layer activations
- Knowledge of model parameters
- Understanding of internal model states

2. Core Principles

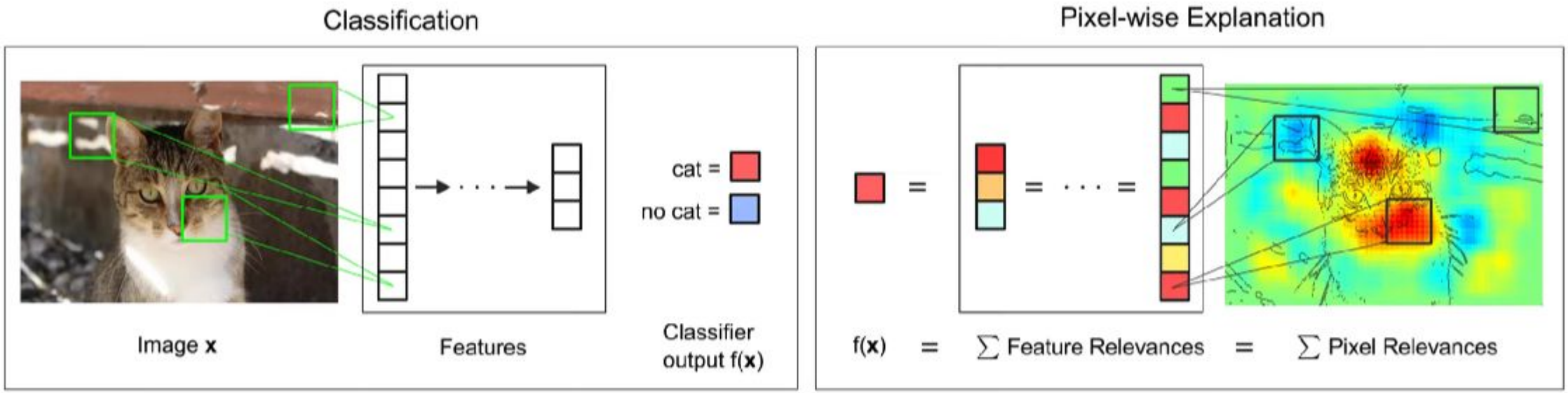
- Utilize backpropagation mechanisms
- Exploit layer-wise connections
- Leverage gradient information
- Consider activation patterns

Major Categories:

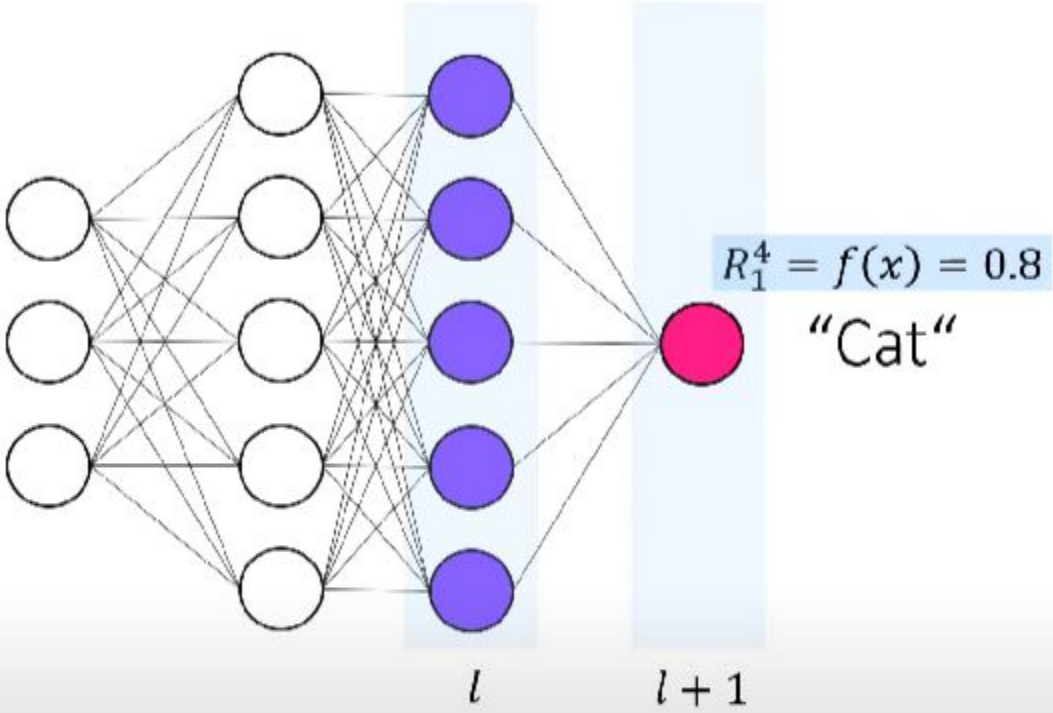
1) Gradient-Based Methods (eg. Saliency Maps),
2) Decomposition Methods (eg. LRP),
3) Attention-Based Methods

Layer-wise Relevance Propagation (LRP)

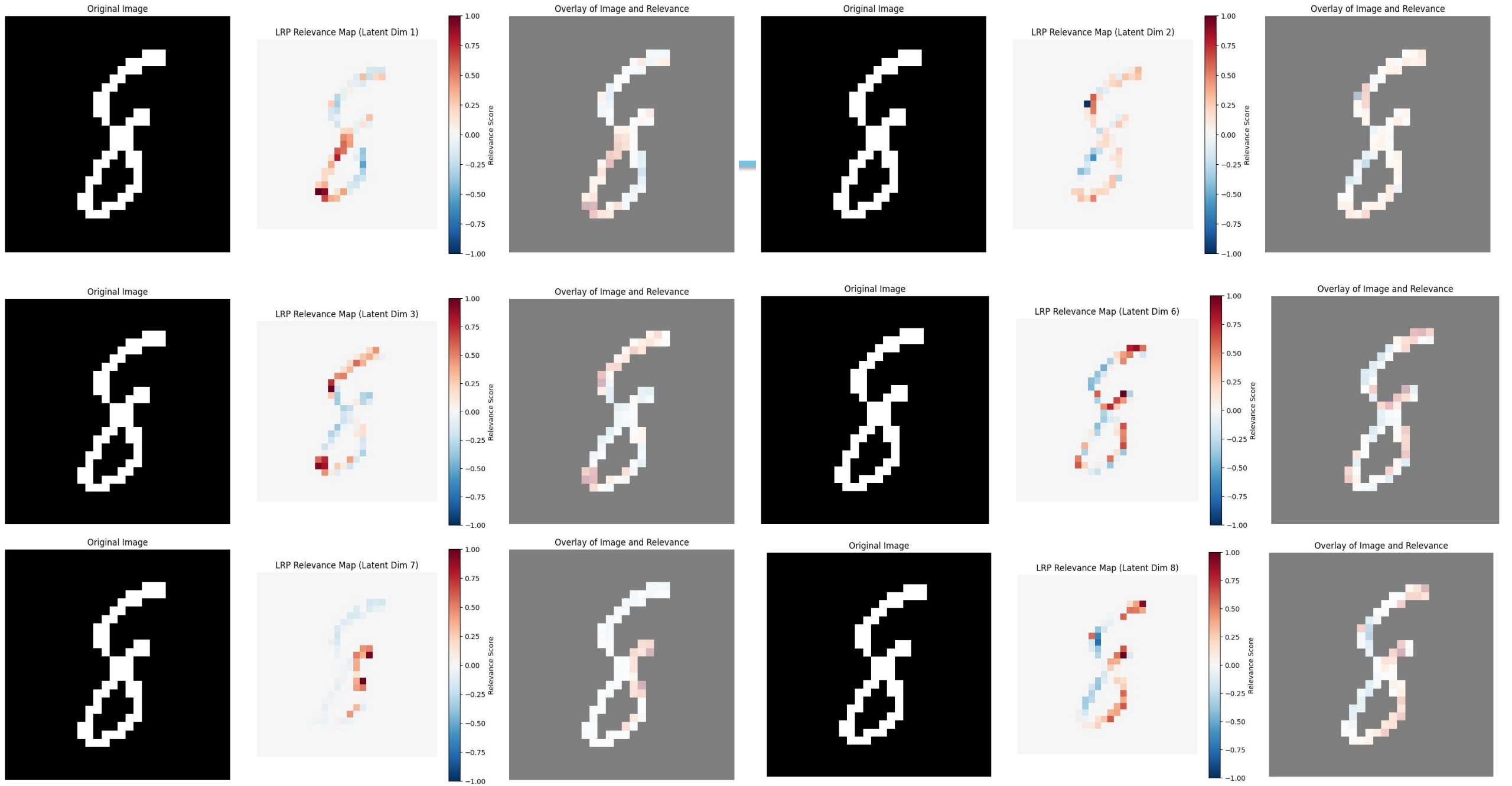
- LRP (Layer-wise Relevance Propagation) is a technique that explains deep neural network decisions by decomposing the prediction backwards through the layers
- It follows a conservation principle where the relevance (importance) is preserved and redistributed layer by layer.
- The sum of relevance scores equals the model's output score, maintaining mathematical consistency

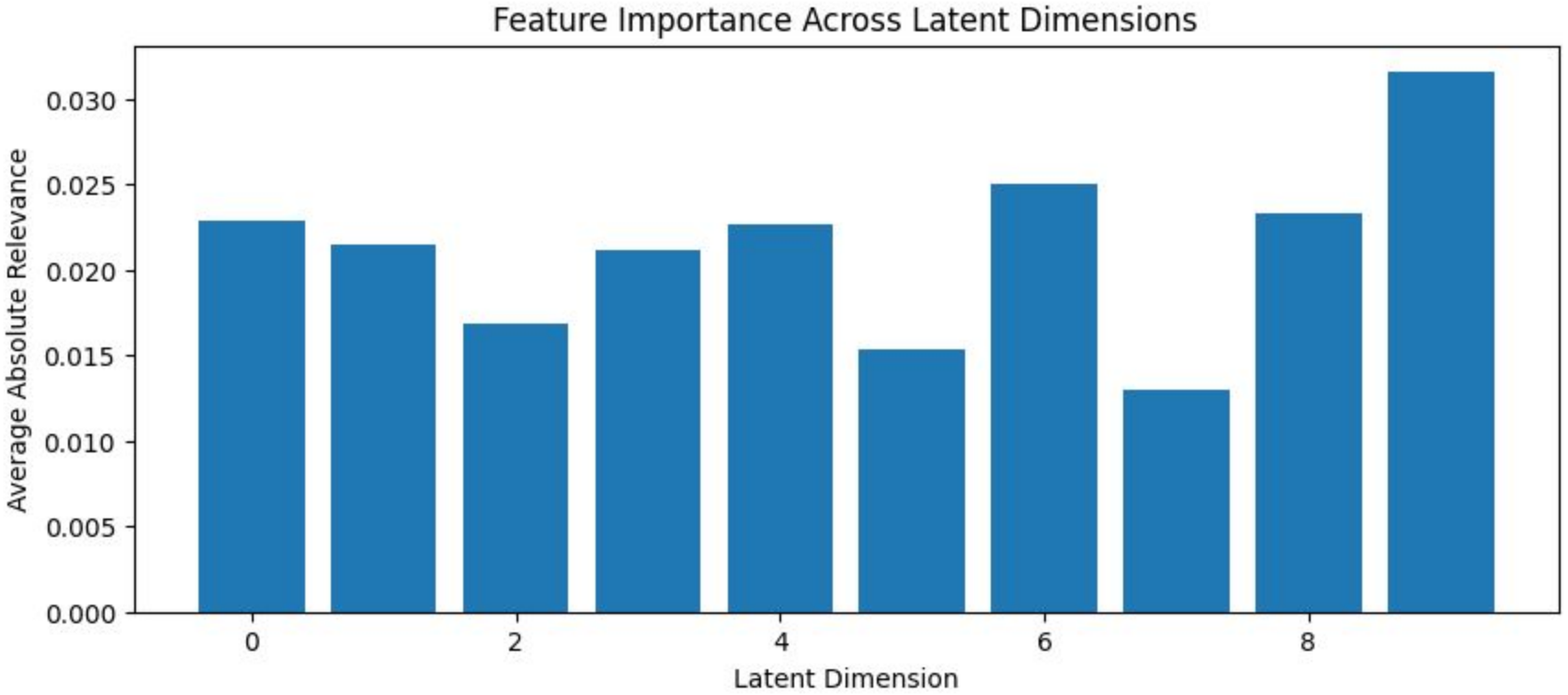


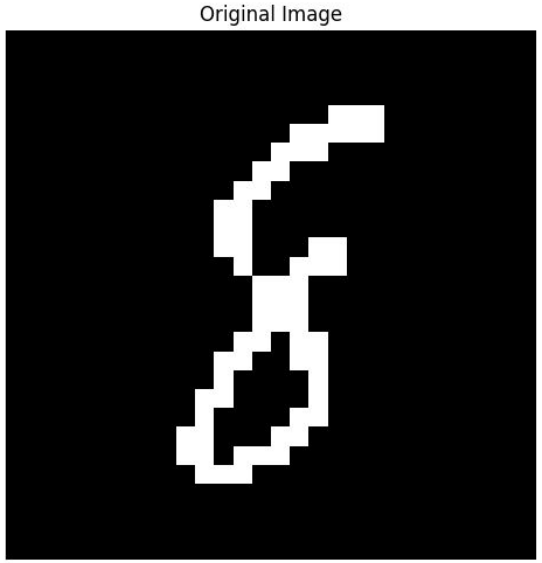
CNN Architecture

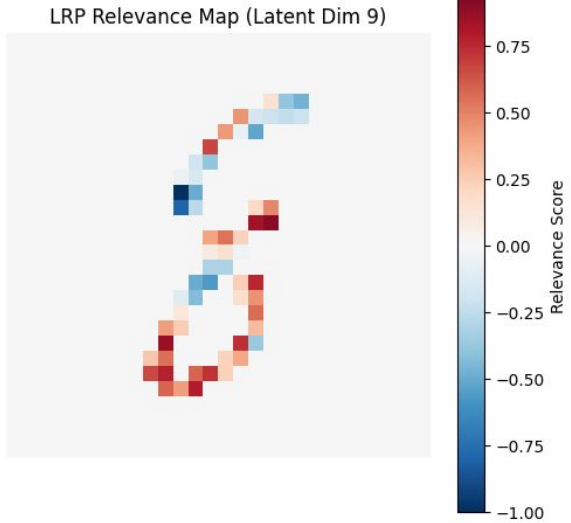


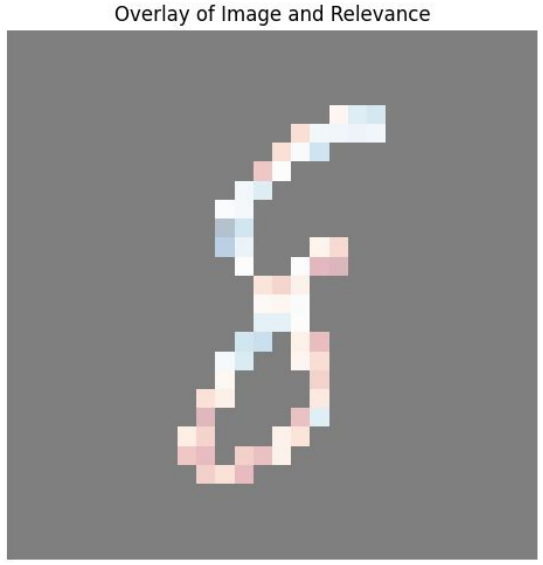
$$R_i^{(l)} = \sum_j \frac{z_{ij}}{\sum_{i'} z_{i'j}} R_j^{(l+1)} \quad \text{with} \quad z_{ij} = x_i^{(l)} w_{ij}^{(l,l+1)}$$


Original Image
LRP Relevance Map (Latent Dim 1)
Relevance Score
Overlay of Image and Relevance
Original Image
LRP Relevance Map (Latent Dim 2)
Relevance Score
Overlay of Image and Relevance
Original Image
LRP Relevance Map (Latent Dim 3)
Relevance Score
Overlay of Image and Relevance
Original Image
LRP Relevance Map (Latent Dim 6)
Relevance Score
Overlay of Image and Relevance
Original Image
LRP Relevance Map (Latent Dim 7)
Relevance Score
Overlay of Image and Relevance
Original Image
LRP Relevance Map (Latent Dim 8)
Relevance Score
Overlay of Image and Relevance
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


Feature Importance Across Latent Dimensions
Average Absolute Relevance
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
2
4
6
8
Latent Dimension


Original Image


LRP Relevance Map (Latent Dim 9)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Relevance Score

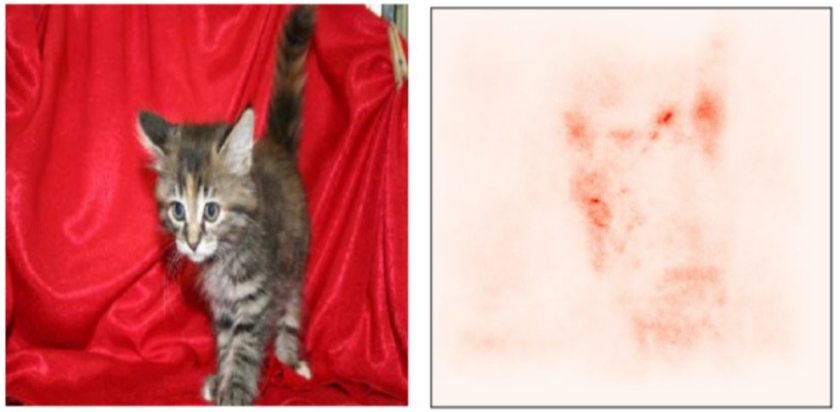

Overlay of Image and Relevance

Saliency maps

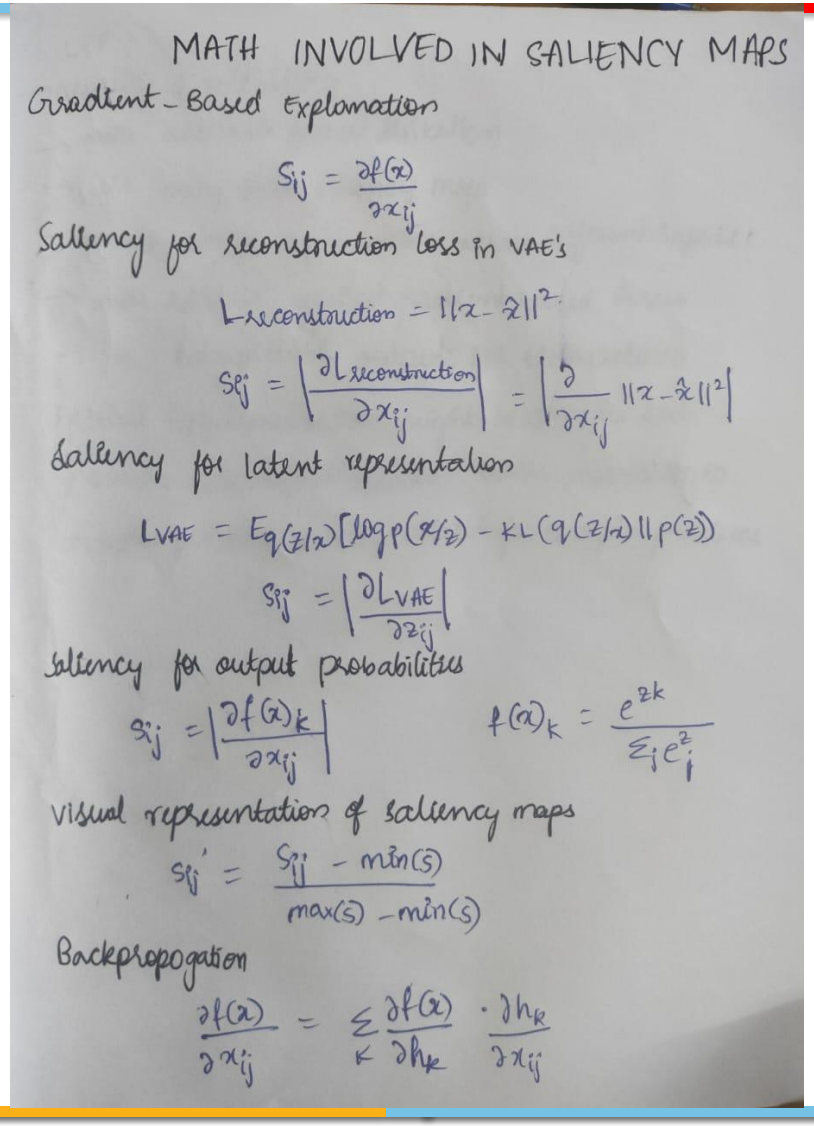
- Saliency maps visualize the important features or regions of an input that influence a model's output.
- They answer the question: "Which parts of the input does the model consider most significant?"
- By highlighting key areas, they make deep learning models more interpretable and transparent.

Applications include:

- Debugging and improving model performance.
- Explaining predictions to build trust in AI systems.
- For image recognition, saliency maps show the regions the model focuses on, such as edges or shapes.
- In this project, saliency maps are used to understand how a Convolutional Variational Autoencoder (CVAE) encodes input features into its latent space.



Math involved



MATH INVOLVED IN SALIENCY MAPS

Gradient-Based Explanation

$$S_{ij} = \frac{\partial f(x)}{\partial x_{ij}}$$

Saliency for reconstruction loss in VAE's

$$L_{reconstruction} = \|x - \hat{x}\|^2$$

$$S_{ij} = \left|\frac{\partial L_{reconstruction}}{\partial x_{ij}}\right| = \left|\frac{\partial}{\partial x_{ij}} \|x - \hat{x}\|^2\right|$$

Saliency for latent representation

$$L_{VAE} = E_{q(z|x)}[\log p(x|z) - KL(q(z|x) \| p(z))$$

$$S_{ij} = \left|\frac{\partial L_{VAE}}{\partial z_{ij}}\right|$$

Saliency for output probabilities

$$S_{ij} = \left|\frac{\partial f(x)_k}{\partial x_{ij}}\right| \qquad f(x)_k = \frac{e^{z_k}}{\Sigma_i e^{z_i}}$$

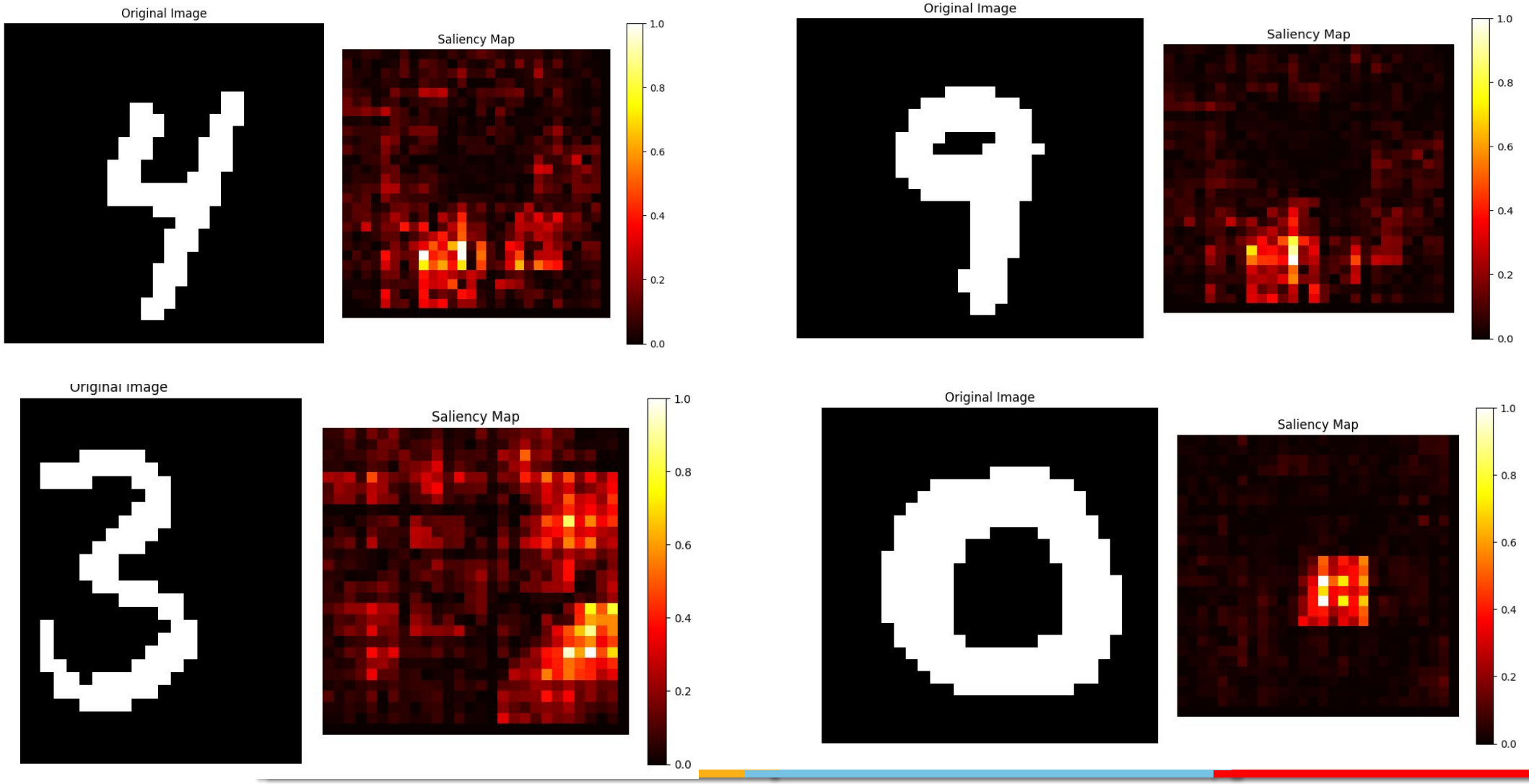
Visual representation of saliency maps

$$S_{ij}' = \frac{S_{ij} - \min(S)}{\max(S) - \min(S)}$$

Backpropogation

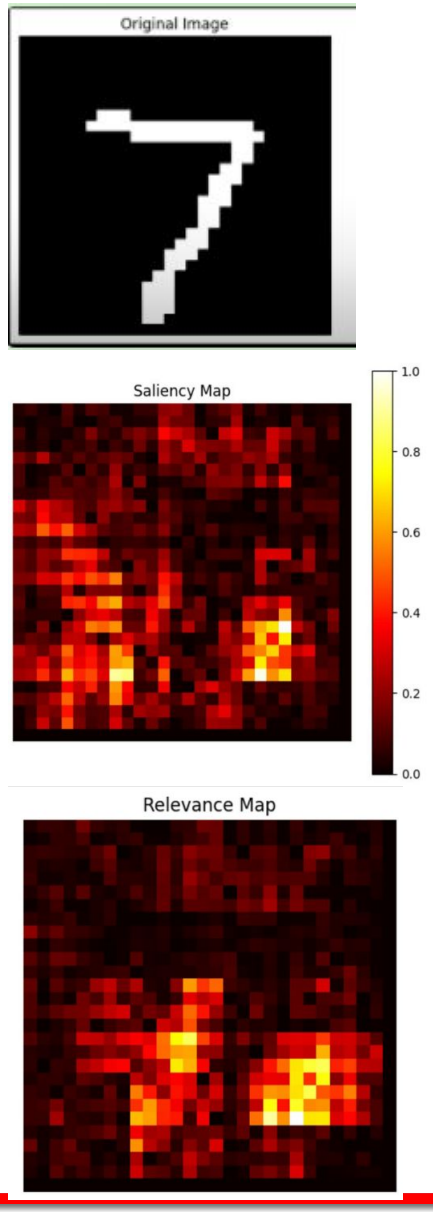
$$\frac{\partial f(x)}{\partial x_{ij}} = \sum_k \frac{\partial f(x)}{\partial h_k} \cdot \frac{\partial h_k}{\partial x_{ij}}$$

Output and conclusion:



Key differences between saliency maps and LRP:

Aspect	Saliency maps	LRP
Definition	Highlights the most influential input features by computing gradients of the output w.r.t. input pixels.	Backpropagates relevance scores through the network layers to explain the model's decision
Gradient Usage	Directly uses gradients to measure input sensitivity.	Does not compute gradients; instead, redistributes relevance layer by layer.
Interpretability	Highlights input features that most affect the output but may not align with human-understandable concepts.	Provides clearer explanations aligned with human reasoning by conserving relevance scores
Strengths	Simple to implement and computationally efficient.	Highlights causal contributions to predictions and provides layer-by-layer insights.
Limitations	Sensitive to model gradients, which can sometimes be unstable and produces noisy results.	Requires specific implementation for different network architectures.More computationally intensive than saliency maps.



Conclusion

Our comprehensive analysis of VAE using multiple XAI techniques revealed complementary insights into the model's behavior. Dimensionality reduction methods (PCA, t-SNE, UMAP) confirmed meaningful structure in the latent space, showing clear organization of digit representations. Neural activation analysis through neuron firing patterns and GradCAM demonstrated specialized network responses to different digit types.

Model-agnostic methods (SHAP, LIME) quantified feature importance and explained individual reconstructions, while model-specific approaches (Saliency Maps, LRP) revealed the internal dynamics of information flow. Together, these methods validated that our VAE effectively learns structured representations of digits, focusing on relevant features for encoding and reconstruction.

Key findings confirmed that the VAE (1) learns meaningful digit representations, (2) shows consistent activation patterns for similar digits, (3) focuses on structurally important pixel regions, and (4) maintains interpretable latent space organization. The combination of multiple XAI approaches provided a robust validation of the VAE's learning capabilities and identified potential areas for optimization.

Thank You