Friedrich-Alexander-Universität
Erlangen-Nürnberg (FAU)

Machine Learning and Perception Group

# Assignment 1: Interpretability with LIME and SHAP

Advanced Topics in Deep Learning

*Muhammad Khalid*
Student IdM: to37gulo
FAU Email: `muhammad.omar@fau.de`

May 4, 2025

# LIME Process Pipeline

The following diagram summarizes the full LIME pipeline for image classification explanation. It begins with the original image, generates superpixels, perturbs them to generate multiple samples, and uses model predictions to train a local interpretable model which highlights the most influential superpixels.

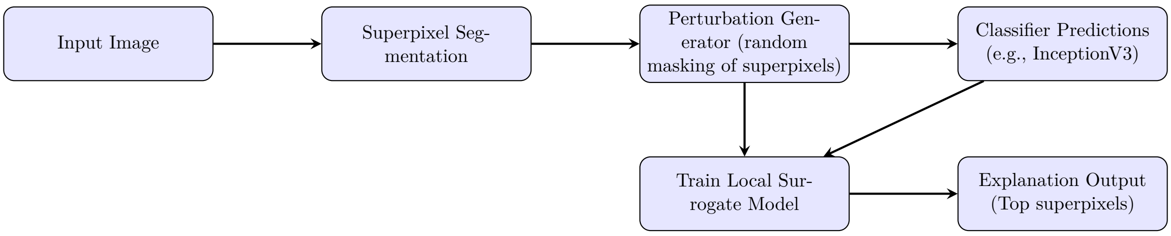
Input Image
Superpixel Segmentation
Perturbation Generator (random masking of superpixels)
Classifier Predictions (e.g., InceptionV3)
Train Local Surrogate Model
Explanation Output (Top superpixels)

Figure 1: Block diagram of LIME explanation pipeline

# Task 1: Local Explanation using LIME

## Objective

This section applies the LIME technique to explain how the InceptionV3 model makes predictions on specific images. LIME builds a local linear approximation of the model by perturbing the image and observing changes in the model output. This helps visualize which regions (superpixels) in the image are most responsible for the model's decision.

## Input Images and Predictions

We begin with three sample images. These were passed through the InceptionV3 model to obtain their top-5 predictions. This step gives us the classification outputs we aim to explain locally using LIME.

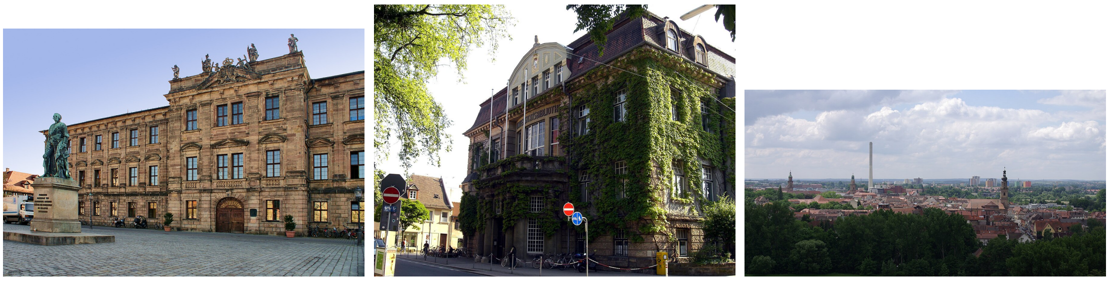

Figure 2: Input images used in LIME explanation

## Superpixel Segmentation

LIME requires segmenting the image into superpixels to serve as interpretable units. We use the Quickshift algorithm to divide each image into meaningful segments that group similar pixels.

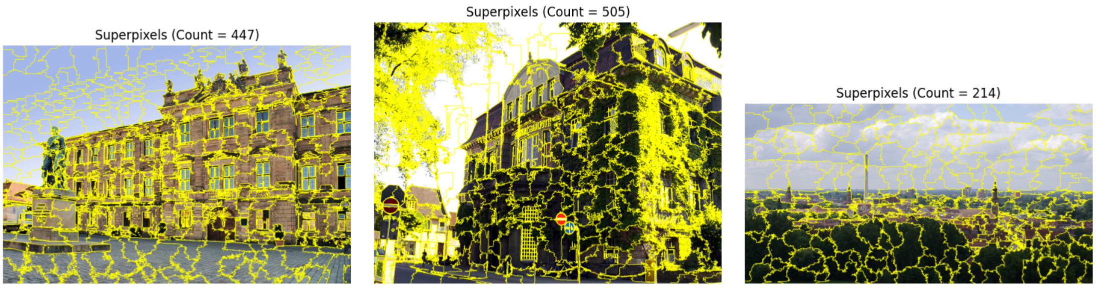
Superpixels (Count = 447)
Superpixels (Count = 505)
Superpixels (Count = 214)

Figure 3: Superpixels extracted for each image

**Discussion:** Superpixels simplify the image while preserving boundaries. They allow LIME to perturb parts of the image meaningfully instead of pixel-wise, which leads to more human-interpretable results.

## Perturbation Samples

Each superpixel is randomly turned on/off across several samples to create a dataset of perturbed images. These are then fed to the classifier to record prediction confidence for the original top-1 class.

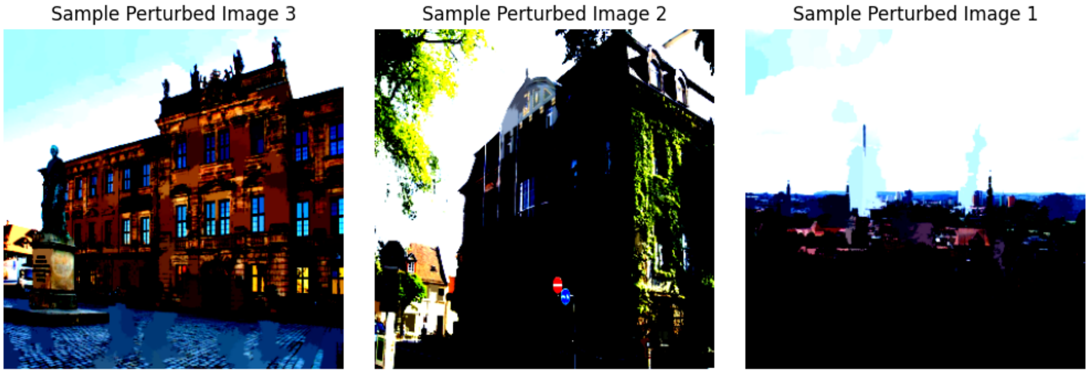

Figure 4: Sample perturbed images used for training surrogate model

**Discussion:** The perturbed samples form the basis of the local dataset on which LIME builds its interpretable model. These variations simulate "what-if" scenarios that show how specific regions affect the prediction.

## LIME Explanation Results

A weighted linear regression model is fit on the perturbed inputs and corresponding confidences. The coefficients indicate the contribution of each superpixel to the final decision.

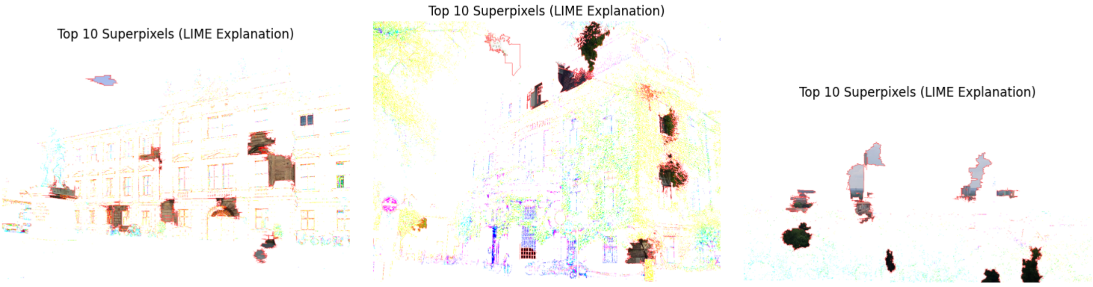
Top 10 Superpixels (LIME Explanation)
Top 10 Superpixels (LIME Explanation)
Top 10 Superpixels (LIME Explanation)

Figure 5: Highlighted top-10 influential superpixels per image

**Discussion:** The results show red-highlighted regions that strongly influenced the prediction. In each case, the highlighted superpixels correspond to visually meaningful parts of the image, indicating the LIME approximation successfully captures local decision logic.

## Task 2: Visual Explanation using SHAP

### Objective

This task aimed to apply the SHAP framework to explain predictions made by ResNet50. SHAP values estimate each feature's contribution to a specific prediction and can be visualized for image data to see pixel influence.

### Computational Constraints

Due to the high computational cost of SHAP on image data and limited resources on both Colab (T4 GPU) and local hardware, full SHAP evaluations could not be completed. Each image took more than 25 minutes to evaluate even with GPU, making it infeasible to analyze all inputs.

### SHAP Implementation Overview

- Model: ResNet50 pretrained on ImageNet
- SHAP Explainer: GradientExplainer with black baseline
- Preprocessing: Resized to 224x224 and normalized
- Samples: Limited to 20 to reduce compute time

**Discussion:** Despite implementation challenges, SHAP remains a theoretically grounded explanation method. Future work can use more powerful servers or optimized SHAP variants to obtain visual outputs for deep networks.

*Refer to the Jupyter notebook for detailed implementation.*