# AI – Assignment 2

Aditya Jhaveri – N13689134

## Anomaly Detection using the MVTec-AD Dataset

- First of all, defined the categories on which work is required to be done. They include leather, tile and grid.
- Downloaded the respective Data modules for each category to train PatchCore models and Efficient AD models keeping in mind the appropriate configurations

### PatchCore

Patchcore uses a pre-trained Wide-Resnet 50 as the backbone as used in the references, number of neighbors are kept as 9 and coreset sampling ratio as 0.1.

Appropriate callbacks of early stopping and model checkpoints are kept for efficient training. After training and testing the model, we find the Image AUROC scores to be 0.5761, 0.6548 and 0.7126, Pixel AUROC scores to be 0.9788, 0.7723 and 0.3626 in the categories: leather, tile and grid respectively. Average Image AUROC score turns out to be 0.6478 and average Pixel AUROC score turns out to be 0.7045 across these three categories.

The anomaly maps, prediction masks and the heat maps are being displayed as follows in the Jupyter notebook.

## Efficient-AD

Efficient-AD is configured to use training and eval batch size of 1.

Appropriate callbacks of early stopping and model checkpoints are kept for efficient training. After training and testing the model, we find the Image AUROC scores to be 0.9297, 0.9473 and 0.9707, Pixel AUROC scores to be 0.9282, 0.7700 and 0.8420 in the categories: leather, tile and grid respectively. Average Image AUROC score turns out to be 0.9492 and average Pixel AUROC score turns out to be 0.8467 across these three categories.

The anomaly maps, prediction masks and the heat maps are being displayed as follows in the Jupyter notebook.

# Similarity Search

To solve this use case, I am following these steps and using Qdrant as advised:

- First of all, sample the data from each category such that 50 belong from the good dataset and 50 belong from the anomaly dataset and combine them to make one collection dataset for one category.
- Make 6 collections, 3 for Patchcore models and 3 for EAD models, each having the name as MODEL-CATEGORY name and hence each containing the data.
- Make a function that extracts features, on the basis of the predictions made by model, extract important features using a pretrained resnet, to reduce the dimensionality to 512 size
- Iterate through all of the categories' images, store them in the respective collections using the respective models and that's how you store feature column vectors in Qdrant.
- To find top 5 similar images, I am making a list of 6 images to query each kind of problem, 2 images from each category: one good and one anomalous

- Iterate through the 6 for Patchore models and visualize the results, same do it for EAD model

Conclusion: EAD turns out to be working effectively than Patchcore, the results in the Jupyter notebook will prove that.

# Patch Description Network (PDN) Receptive Field Calculation

The following proof and conclusion in written as a markdown code block in the jupyter notebook submitted, please refer to that.

Note: All the visuals regarding to any task have been already been loaded in the Jupyter notebook. All the required knowledge about the code statements have also been documented and commented properly in the Jupyter notebook. If any confusion, please let me know.