Brief overview of the problem:

The pursuit of image fidelity is one of the most important concerns in the realm of graphics and digital imaging. We continuously seek ways of enhancing the visual quality of images that offer users a more immersive and realistic user experience.

Problem:

The rendering process converts 3D scenes into a 2D image output. The challenges of noise and low resolution often plague rendered images which is highly undesirable. They can arise due to limitations in computational resources, rendering algorithms, or noise in the scene. Traditional image processing techniques exist that address both of these issues like interpolation, anti-aliasing, and filtering. However, machine learning offers a powerful way to address these challenges post-rendering. They form a part of the image enhancement networks in the realm of neural rendering.

1. What data would you use?

   a) Super Resolution:
   Consists of low resolution images typically obtained by resizing high resolution images, capturing images with low res sensors or compressed images. During training, the models are trained on pairs of low resolution and high resolution images to learn the mapping between them. During testing, the trained model takes a low-resolution image and generates a high-resolution version.

   b) Denoising:
   Similarly, for denoising, the input data consists of noisy images. During training, the models are trained on pairs of noisy and clean images. The model learns to identify noise and also remove it while preserving the image. During testing, noisy images are fed to the model to produce clean versions.

2. What are your key input and output variables? Review the following sections:

   a) Super resolution:
      i. **Input variables**: Low resolution image i.e. the input image that needs to be enhanced. Could be RGB or grayscale.
      ii. **Output variables**: High resolution image i.e. of the output of the super resolution process. The output variable would replicate the input channels whether RGB(3) or grayscale.

   b) Denoising:
      i. **Input variables**: Noisy image i.e. the input image containing the unwanted noise that needs to be removed.
      ii. **Output variables**: Denoised image i.e. the output of the denoising process which produces a clean version of the noisy image.

3. What type of machine learning problem is this?

   Both of these problems are similar.

   Super Resolution/ Denoising

   a) Regression/Classification: This is a ***regression*** problem. Output is an image which is continuous.
   b) Supervised/Unsupervised: This is ***supervised*** learning problem. Pairs of low-res/high-res or noisy/denoised images are provided at the time of training.
   c) Parametric/Non-parametric: CNN's are ***non-parametric*** models. They require a lot of data to be trained and make no assumptions about the shape of the function to be estimated. They do have a fixed number of trainable parameters.

4. What steps would you take to solve this problem through machine learning?

   10 steps for a machine learning problem:

   a) Define the purpose of the ML project: The challenges of noise and low resolution often plague rendered images which is highly undesirable. Super resolution and denoising in rendering offer a powerful way of improving the visual quality of images for a better user experience.
   b) Obtain the data set for the analysis: There are existing datasets that exist (DIV2K) for super resolution. Often pairs of low-res, high res can be created by downscaling the high res images. Similarly, noise can be added to clean images for generating pairs of noisy, clean images using image processing tools like OpenCV.
   c) Explore, clean and preprocess the data: Data augmentation can be done by adding rotations and flips. Also, images can be preprocessed by considering – image cropping, scaling and normalization which helps the model learn better.
   d) Dimension reduction and feature engineering: This is automatically done as this is a deep learning based approach.
   e) Determine the ML task at hand: As describe above, this is a *regression, supervised and parametric* problem.
   f) Partition the data (if supervised ML): Since this is a supervised problem, the image dataset obtained in step 1 will be split into training, test and validation sets.
   g) Choose the ML technique: Deep learning techniques are commonly used.
      i. Super-resolution: CNN's/GAN's
      ii. Denoising: CNN's/Autoencoders
   h) Use the ML technique: Apply the CNN 's to generate relevant outputs

i) **Interpret the results**: The results can be inspected visually or using metrics like signal to noise ratio or structural similarity index.
j) **Deploy the ML technique**: Once the models are constructed, integrate them in the rendering pipeline.

5. **What might cause missing data in your data set? Which approach outlined in the lecture materials do you think would be most suitable for dealing with missing data, and why?**

Missing data in context of super resolution and denoising can be due to various factors.

a) **Super resolution**: If the input images are very low resolution, it doesn't make sense to downsample it further as the model won't learn the proper mapping from low res to high res. Data can also be corrupted and lack significant variation of resolution.
b) **Denoising**: If the dataset has insufficient noise in the input data, it will fail to generalize well or if the noise model used to add noise to data doesn't capture the noise characteristics, the model here won't effectively learn the mapping between a noisy and clean image.

In both cases, it would be best to exclude images that are too low resolution in case of a) and for b) best, to include only images with noise levels that will help the model learn. Data imputation or remove the effected records is the way to go in this case.