I have been working as a software engineer with Adobe in computer graphics for the past few years. Specifically, I have been a part of a team that develops software for photorealistic 3D rendering. It is a virtual staging software that allows you to import 3D assets, then arrange and stage them to build a scene. Users can add lights to further customize the appearance of the scene. Commonly, virtual staging software uses a camera-based system to allow users to set up shots for rendering. The rendering techniques used are mixed - both real-time and ray-tracing rendering is supported.

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, I believe 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.

**Why machine learning is a good fit:**

* Super Resolution: It refers to the process of enhancing the quality and resolution of an image beyond its original dimensions. Traditional interpolation methods like bicubic interpolation *usually fall short in preserving sharp details leading to unsatisfactory results*. Machine learning based super resolution is a technique that leverages deep neural networks to predict high-resolution details from low-resolution outputs. CNN’s are the workhorses of image-related tasks and can easily be used for this process. However, GANs can also be used for the same. SRCNN is the pioneering architecture for super-resolution CNN’s.
* Denoising: Noise is often a part of rendered images, and can be removed through denoising techniques. Machine learning based denoising models can be designed to remove noise using unsupervised machine learning techniques where noise can be added to training set to trick the neural network into learning the mapping between noisy and denoised image. Autoencoders and CNN’s can both be used for this particular task. DnCNN has been used recently for its effectiveness.

**Advantages over traditional image processing techniques:**

* Capacity to learn complex hierarchical representations from data.
* Feature learning – In the case of super-resolution, high-frequency features that contribute to image quality are learned and in denoising, the difference between signal and noise is learned.
* Residual connections help share knowledge between original and downsampled or noisy data.

**Training Set:**

In both these cases, the network is trained on a large dataset of image patches, consisting of pairs of low-resolution and high-resolution patches for super-resolution and noisy and denoised images for denoising. The network learns to minimize the difference between the predicted high-resolution/denoised patches and the ground truth high-resolution/denoised patches in the training set. By integrating both approaches into a unified pipeline, the resulting images benefit from enhanced resolution, reduced noise, and overall heightened visual quality.

**Challenges:**

While the impact of machine learning in solving both of these problems is transformative, challenges certainly exist. Optimizing models for real-time application, treating any artifacts that might be introduced during enhancement, and ensuring generalization to diverse scenes are some common issues that will need to be addressed.

**Questions:**

* How seamlessly can these two post processes be integrated into the rendering pipeline?
* What are the computational resources associated with implementing denoising and super-resolution after rendering?
* How to ensure enough variability in the training data so that the model can generalize well?
* Does it impact the realism of the rendered image?
* Is it possible these processes might introduce new artifacts?
* What metrics to use to evaluate image quality?

**References:**

[1] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang- Image Super-Resolution Using Deep Convolutional Networks

 [2] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, Lei Zhang - Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising