

Question 5

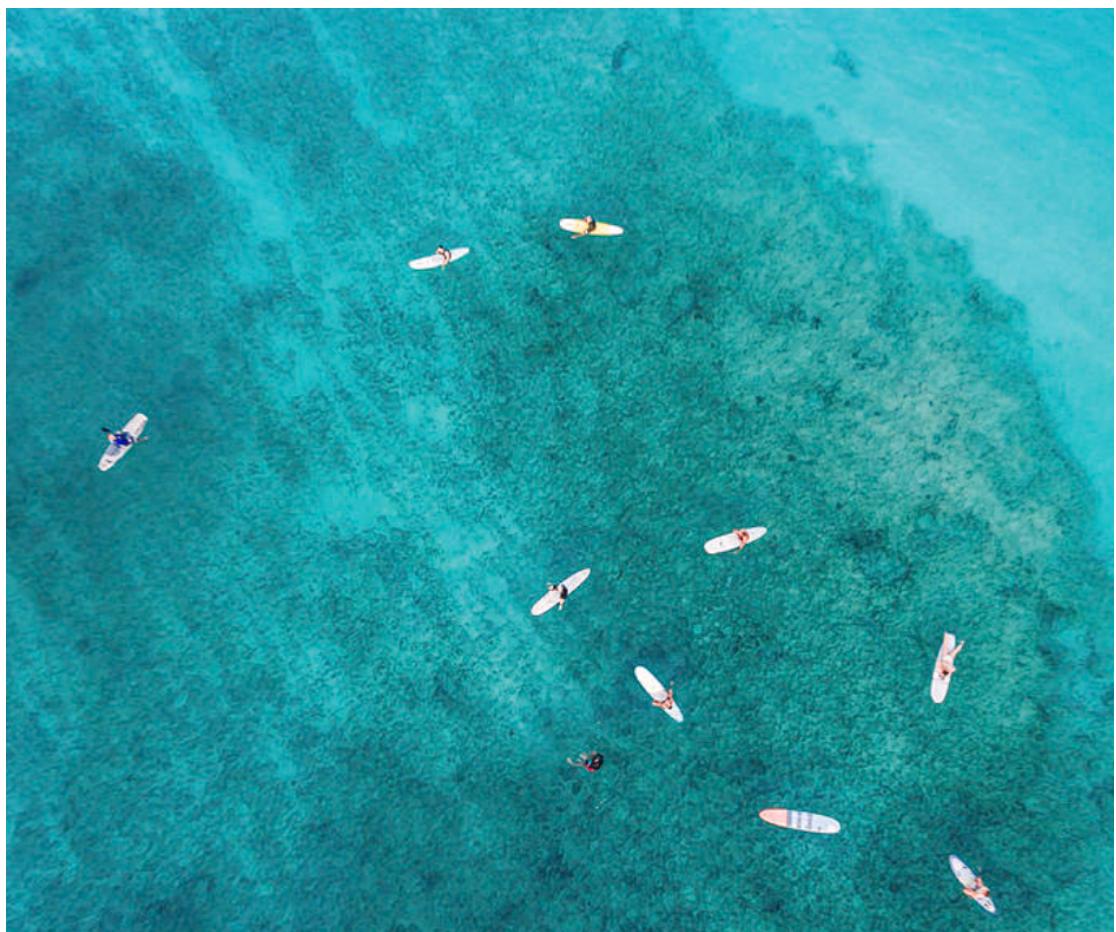
Donkey image after texture synthesis has been used to remove the donkey:

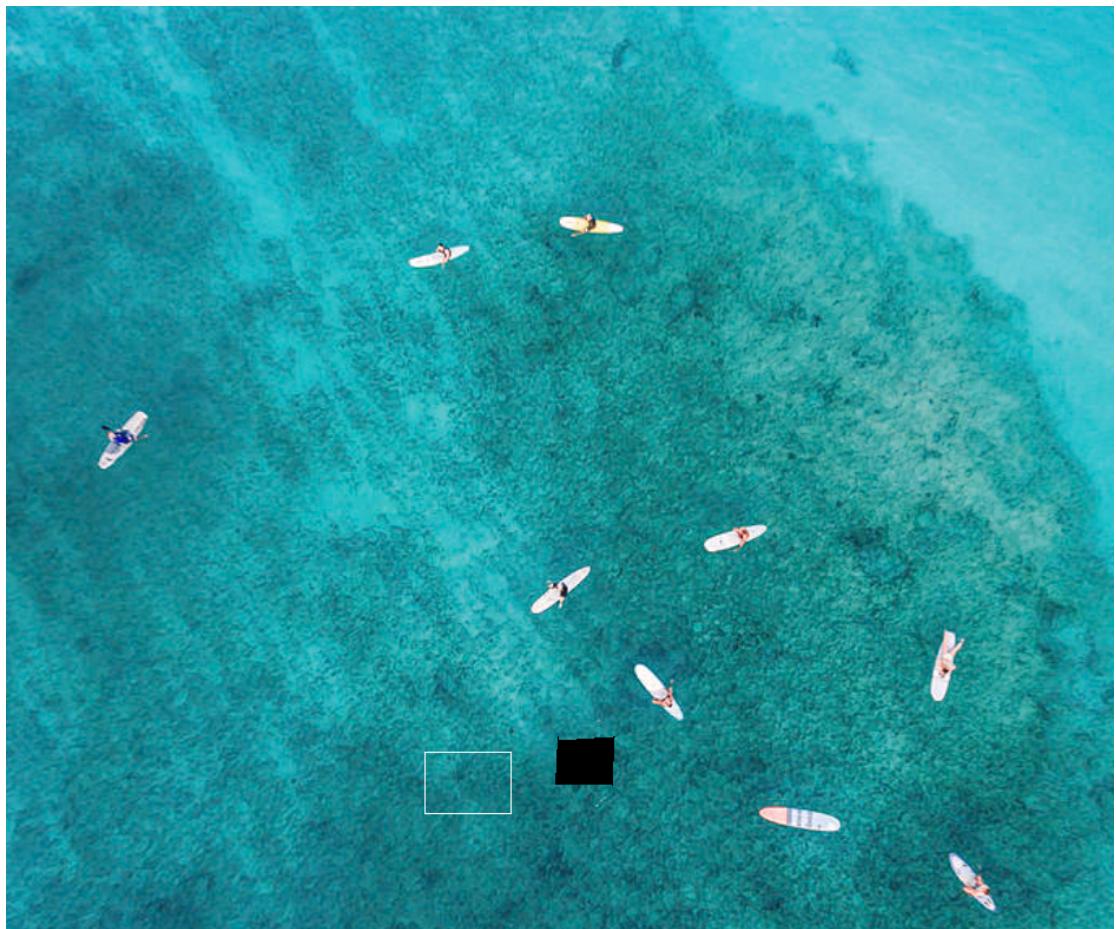


Question 6

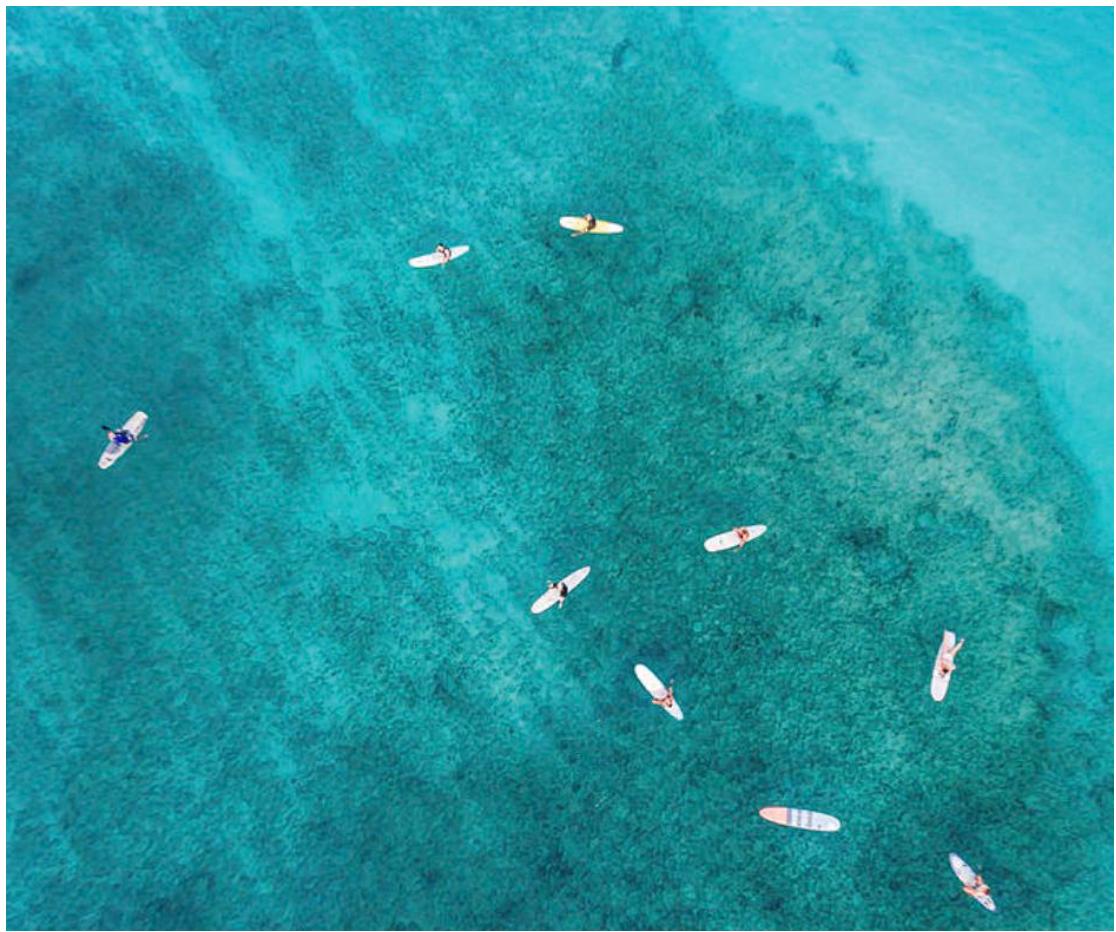
Example 1 (performing well)

Original image:





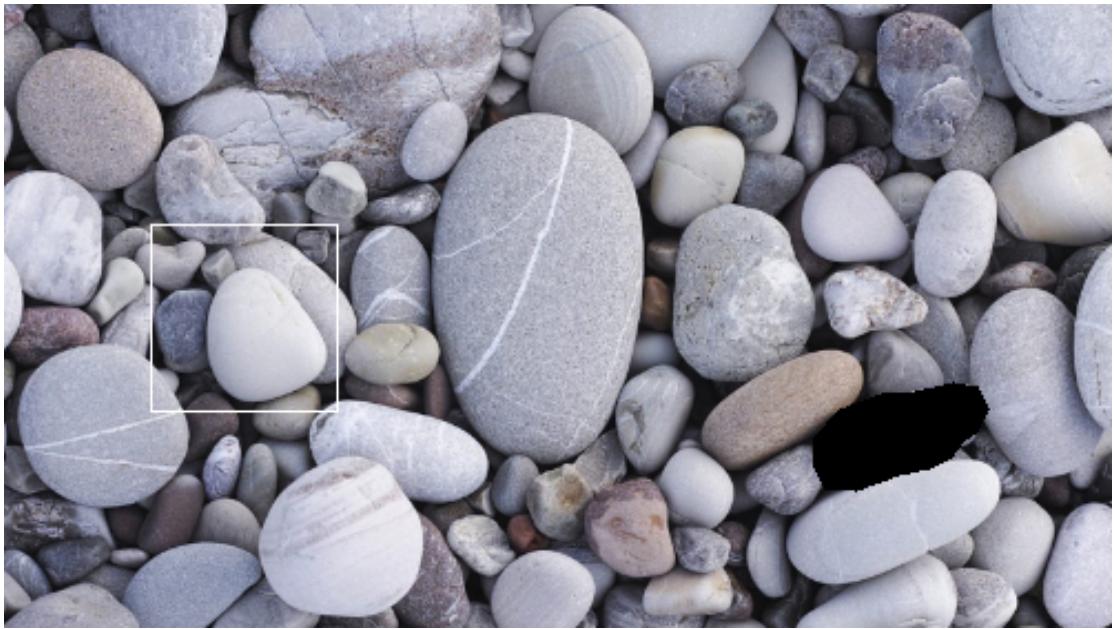
Modified image:



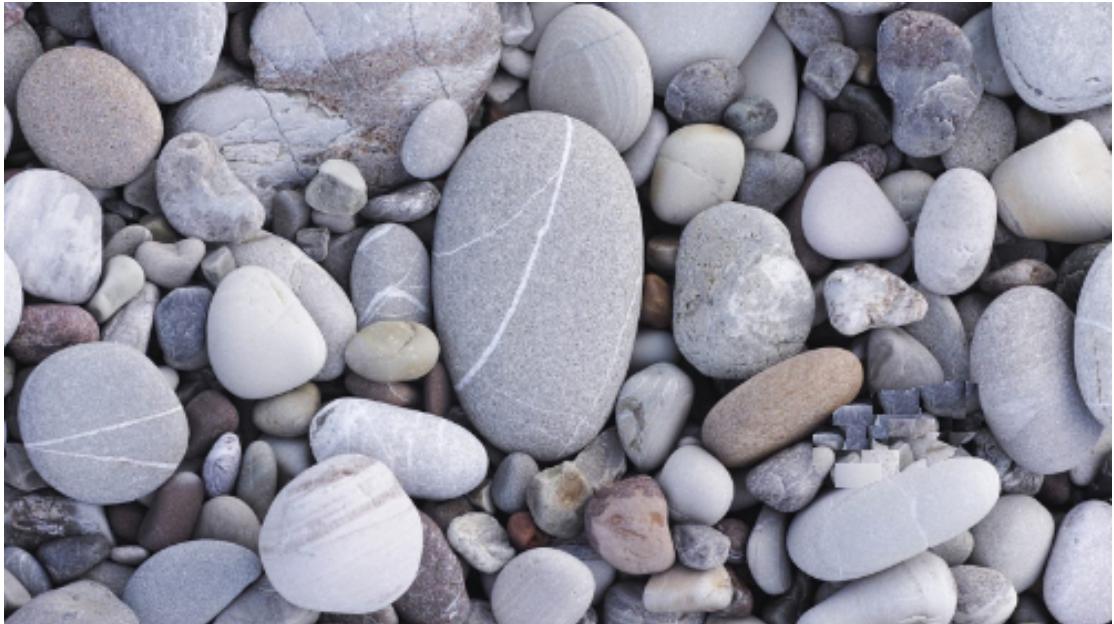
Example 2 (performing poorly)

Original image:





Modified image:



The algorithm does not perform well in the pebbles image in which I removed a white pebble. The algorithm performs poorly on this image because the texture of the pebbles is not repetitive. The texture of each pebble is so distinct that the algorithm could not capture the global patterns with small patch size, and it does not find good matches to fill the hole. In addition, the pebbles with different colors, sizes, shapes, and orientations makes any misalignment conspicuous.

Question 7

The randomPatchSD parameter has effects on deciding which patch to sample from. If it is too small, the selected sample patch is the best match one, which is deterministic. It is very likely that the result contains excessive repetition. If randomPatchSD is too large, we will have too many patches to randomly sample from. This will cause poor

matches that will mess up the texture at a local level, leading to unrealistic textures.

The patchL parameter determines the size of the patches. When it is too small, the algorithm may focus too much on local details of the input texture and fail to capture global patterns and structures. As a result, the synthesized texture may lack coherence and look inconsistent. If it is too large, there may be repeated patterns in the output texture since the same patches may be selected many times due to a limited number of available patches. In addition, larger patches tend to average out the local variations in the input texture and generate an overly blurred output, and thus resulting in the loss of finer details in the synthesized texture.