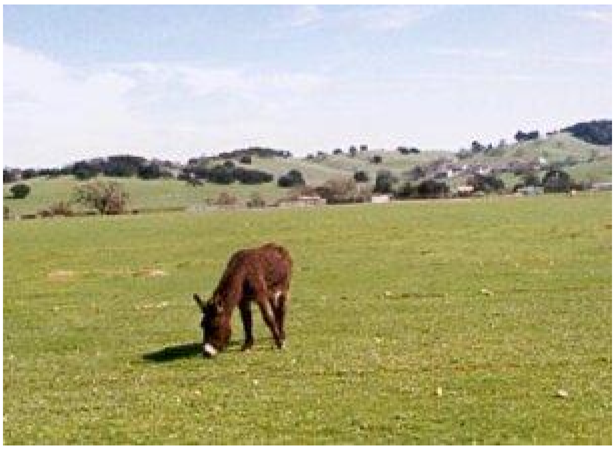
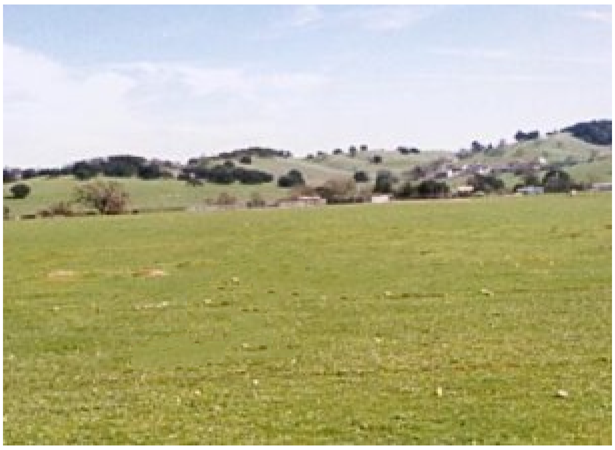
CPSC 425 | Assignment 3: Texture Synthesis (Efros & Leung) | James Graham 10571446

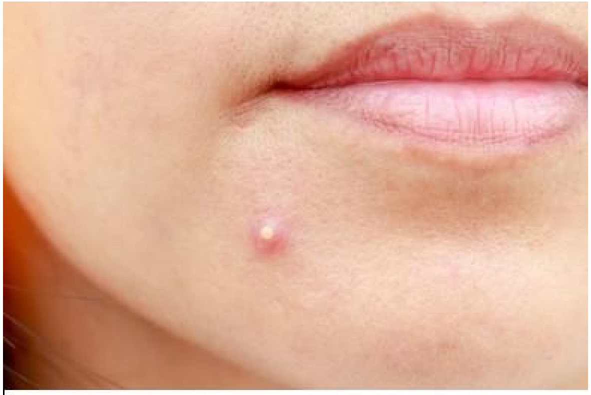
Donkey before



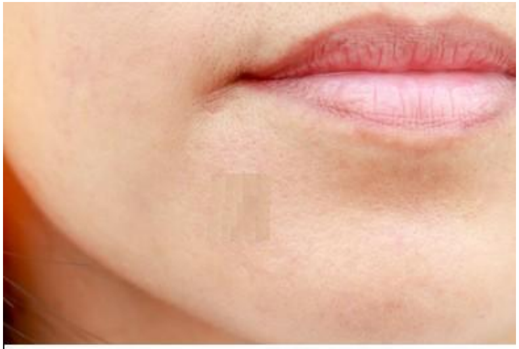
Donkey after



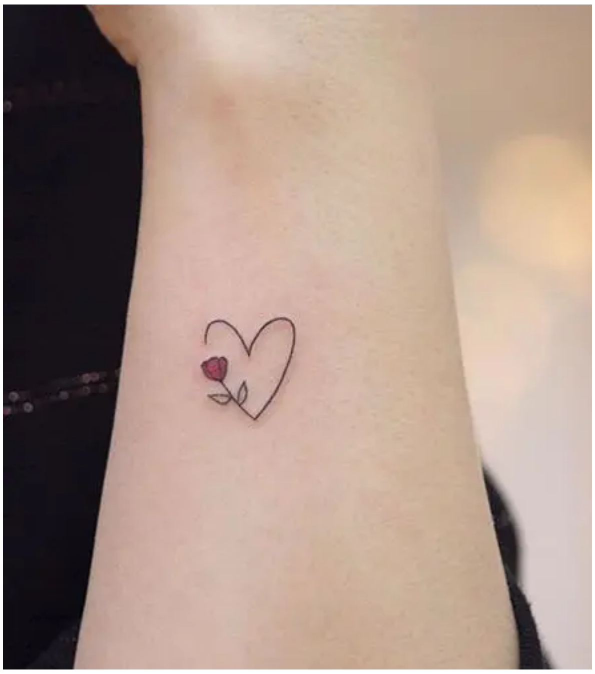
Pimple before



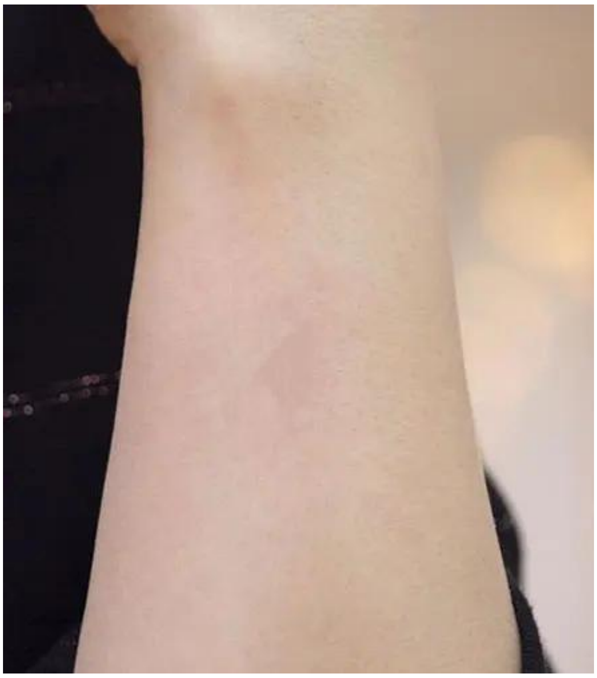
Pimple after (bad photo) - sd = 10, patchL = 10



Tattoo before



Tattoo after (good photo) - sd = 0, patchL = 2



6. Briefly describe why the method failed in the case in which it performed poorly.

The image that performed poorly was the pimple image, which had a random standard deviation set to 10 and patchL to 10. These changes can be easily seen in the after image where the filled region has a blocky appearance and colours that don't match the skin. The reason for the blocky appearance is because of the patchL size being set to the large size of 10, mixed with the large random standard deviation, which results in colours that aren't always the most optimal in terms
of matching pixels around the hole.

7. Provide an explanation for the effects of the randomPatchSD and patchL parameters. What results can be expected if these values are too small or too large, and why do these results happen?

randomPatchSD is the integer value that helps determine the possible colour range of each patch that fills in the hole. It provides the algorithm with a hint of randomness so that all of the patches are not the same colour (which wouldn't look realistic).

patchL is the integer value that represents the size (patchL*2+1 x patchL*2+1) of the patches that fill the hole.

When randomPatchSD is too small (e.g if randomPatchSD = 0, the minimum of the ssd image is always chosen and this gives the optimal patch), it's more likely to provide a successful fill when the hole to be filled is a similar colour and tone to the area selected to fill the hole. This is because the factor of randomness between the patches is minimized. The fill is not always good when we choose a small randomPatchSD value though, for example, if we select an area to fill the hole that isn't similar in colour or tone to the hole it won't adjust for this discrepancy. It really just forces the patches to have similar colours and tones.

When randomPatchSD is too large, we're likely to see larger discrepancies between the patch colours that filled the hole. This is because the range of random colour values that are possible are greater. In this case we can clearly see the different patches used to fill the hole because of their difference in colour.

The smaller patchL is, the longer the runtime, and more smooth-looking the final filled result is. The resulting patched area appears smoother because there are a lot of small patches being used to fill the hole, also leading to a longer runtime because there are so many patches needing to be used if the size is small.

Larger patchL values result in quicker runtimes, and blocky-er hole-fills. Vice versa to the smaller patch size, the larger patch size will quickly fill the hole and will shorten the runtime, but because of the larger patches there may be large outlines clearly visible between the patches being used to fill the hole (this also depends on the randomPatchSD value of course).