Patching Holes

16:198:461

Consider the following images, **Leaves_Masked.jpg** and **Wood_Masked.jpg**:

Figure 1: Two images, 900 x 900 pixels, with the middle 1/9th removed.

In each of these images, the middle 300x300 pixels have been removed (rather, have been painted white). The masked files are available with this PDF. The goal of this final project is to build a machine learning system capable of predicting or filling in these pixels, to 'realistically' complete the image. There are many ways this might be approached, for example trying to find a relationship between related or nearby pixels, and using that to fill in missing pixels.

$$(\textbf{surrounding visual context data}) \mapsto (\textbf{missing visual data}) \tag{1}$$

In building a system to accomplish this, you need to consider the following elements:

- **Input Space:** What are you taking as input? How is it formatted and structured? Notice that form image data, spatial relationships matter. Is your input numerical or categorial? What training data do you have available to use?
- **Output Space:** What are you taking as output? Do you want to predict one pixel at a time, or multiple pixels, or the entire patch? Does your output need to be numerical, or categorial? What training data do you have available to use?
- **Model Space:** For your choice of input space and output space, you need to consider a model that maps between them. What kind of models make sense? What kind of models are easier or harder to work with? Will your model be expressible enough to capture the relationship between input and output? How do you know if your model is too expressible?
- **Loss:** What makes one model better than another? Notice that in this problem the quality of a model is perceptual / aesthetic - how nice does the result look? But we need a numerical measure of loss to make any of our tools work.
- **Training Algorithm:** For your choice of loss function, you need an algorithm to produce a good model. What is applicable to your choice of model, loss, and data? What are potential problems to consider, or choices you have to make for your algorithm?

For the final project, you need to develop two systems to solve the above problem - one based on trees, the other based on something other than trees. For each system, you should have a writeup that covers each of the five parts above, describing each part for your system and outlining your design choices, the problems you faced and how you solved them, and summarizing your results. For each system, you should be able to present as a final result a 'completed' **Leaves** and **Wood** image, and assess which approach was more successful.

Your report should at least cover the following:

- **Training and Testing Data:** For this project, you're welcome to develop your own testing and training data sets, but be explicit as to what you are doing and where these images are coming from. You might, as I did, go outside and take some pictures of trees, and use this to develop a training data set. But be clear in your report what you did and how you used the data you collected. Depending on how big your model is and what it maps between, you may actually be able to harvest some useful training data from the masked images themselves. Note that we can even apply data augmentation here, as the same image rotated or flipped may provide novel training information as well!

- **Structure of Data:** In class, we focused on data structured as vectors of features, $\underline{x} = (x_1, x_2, \ldots, x_D)$. How should you structure your data here? What features are relevant to consider at any point? Notice that the bigger each data point is, the bigger your model may need to be, and the more data over all you may need to be successful. Additionally, note the structure of the data may dictate what the model can do. For a given missing pixel, do you want to fill it in using predictions based on pixels above it? Below it? To the side? All three?

- **Data Preprocessing:** As we've discussed a few times, the 'raw' data may not be the best thing to base your model off. What kind of pre-processing might be useful to clean or structure or organize your data, to improve the results of the final model? Note, as an example, that colors can be viewed as numerical (e.g., RGB values), or categorical (e.g., 'red').

- **Model, Models, Modeling:** Within the constraints that one of your systems be tree based and one of your systems be non-tree based, you have a lot of flexibility. Do you want to treat this as a classification problem? A regression problem? Some kind of combination? Do you want to build a single model to do the task, or do you want to try to combine a collection of models? Note that the more complicated the model, the more data you need to train it (and the more time you may need as well!), so be mindful of the tradeoffs here - not to mention the risk of overfitting.

- **Bootstrapping:** One opportunity that this problem introduces is *bootstrapping* - when you fill in some missing data or pixels, you can use these predictions to help fill in *more* missing data or pixels. This can be a very powerful tool - but also potentially introduces risks, as you would be making predictions based on the results of other predictions. How can you use bootstrapping, and how can you be mindful of or account for the potential risks involved?

- **Measuring Loss:** Obviously a good model is going to fill in the missing parts of the image 'realistically' - but how can we measure that? We would want the predicted data to be 'close to' the actual data, but what does that mean for this problem? Does your numerical loss value actually correspond to visual quality?

- **Training:** Obviously the goal in training is to build a model that performs well on the training data and generalizes well to non-training data as well. How do you approach this, for your model? Is overfitting an issue? If so - how can you try to minimize overfitting? If not - are you sure that your model is expressive enough to really capture the data? Note that time and efficiency will increasingly be an issue as we approach the end of the semester.

- **Assessing Your Models:** Obviously the final 'goal' is to fill in the Leaves and Wood images. But since you don't know what the original images looked like, you can't use these to assess the quality of your model beyond visual appeal. So in addition to training your models to achieve low loss, how can you verify your model is accomplishing the desired task?

IMPORTANT NOTE: Your final grade will *not* be based on the visual quality of your final results - only on the thought, analysis, and development of your solutions.

For this project, when it comes to machine learning models and algorithms, you need to implement your own solutions. For things outside the purview of the class (such as image file reading/writing, graphs, visualization, data structures), you are welcome to use outside libraries and sources (though obviously cite your work and be clear when you do). There is no language requirement, so choose one you are comfortable with, that is able to handle the desired tasks.

Bonus 1: If you are inclined, spend some time learning a machine learning library or framework, and implement a solution to the problem in it. How does its performance compare with your own?

Bonus 2: If you are inclined, look up developed inpainting solutions (for instance with Stable Diffusion), and use one to solve the problem. How does it compare to your results?