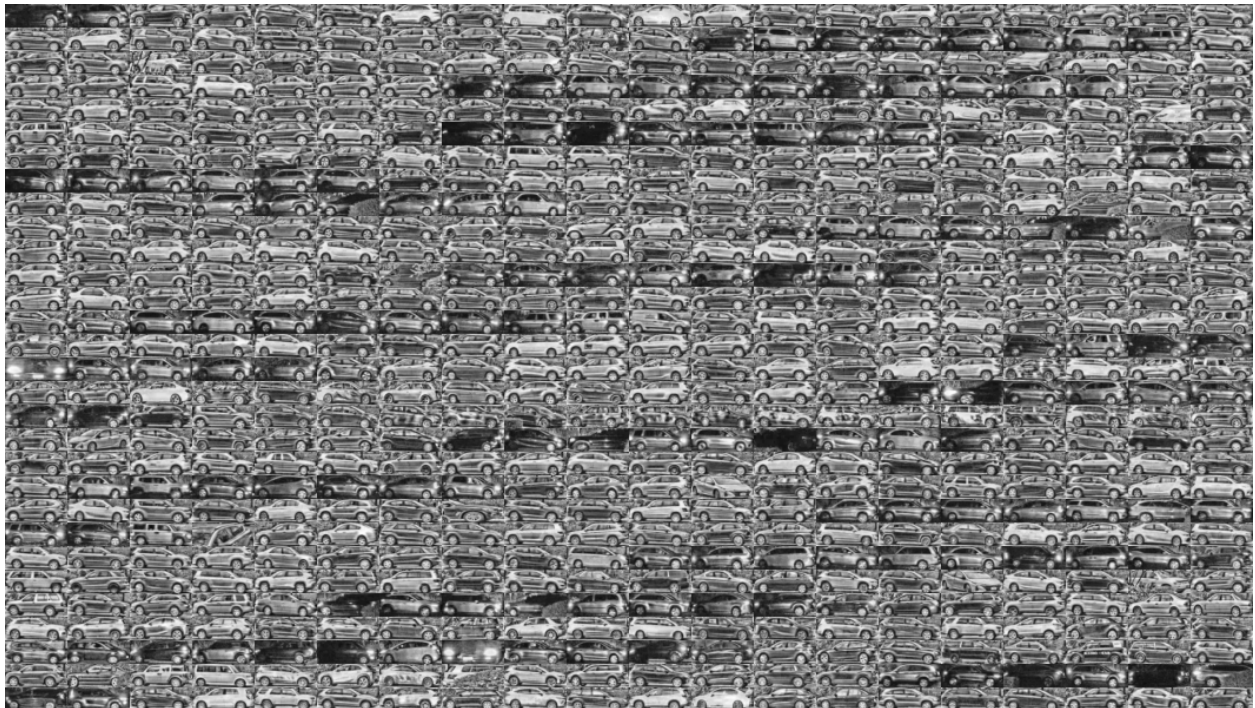


Exploring with UMAP

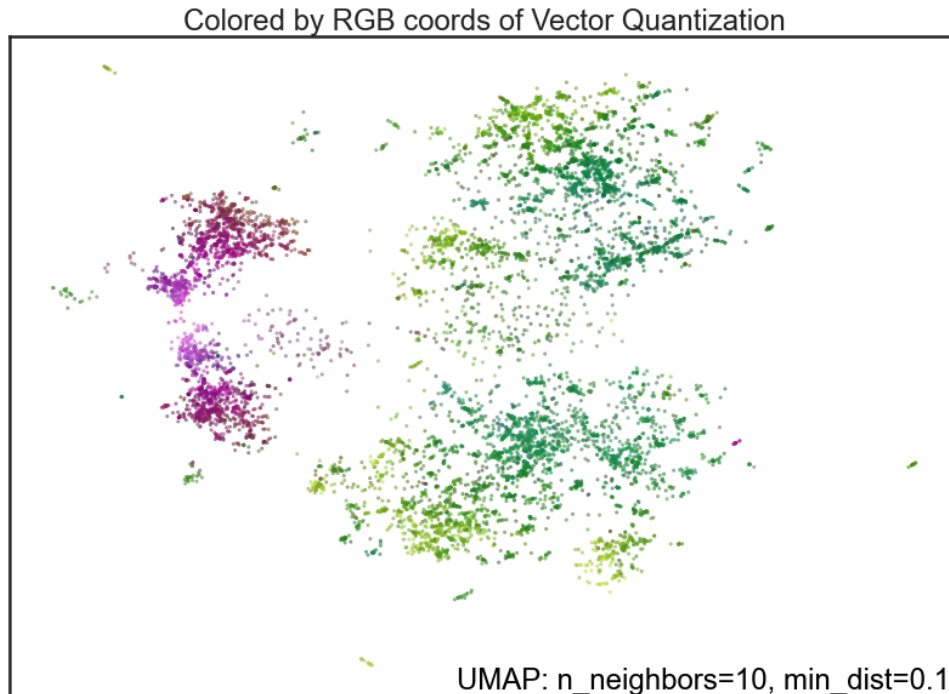
J.Beale Dec. 2021

I'm impressed by the general sorting ability of a program called UMAP. It is used for studying many kinds of data sets. UMAP groups sets of numbers by similarity, which sounds less impressive than I think it is. It is *completely* general-purpose, by which I mean it makes no assumptions at all about what the numbers represent. It works on anything which can be described somehow by sets of numbers.

To take an example, I have a set of images from a roadside camera, which were marked as containing cars by an object-recognition program called YoloX. Some taken during the day, some at night, mostly from the side but a few at other angles. YoloX makes a rectangular crop of each detected car object. I converted that to grayscale, and scaled it down to a small fixed size of 53 x 20 pixels. Below is a sample of 600 such images in my data set.



To convert these images to numbers, I reshaped each 53 x 20 pixel grayscale image into a linear array of $(53 \times 20) = 1060$ numbers, each number being the brightness of one pixel. You might think that discards some information. UMAP sees that 1060 element array as a single point in 1060-dimensional (!) space. I then fed 10,000 such points (from 10k different images) and asked it to plot that entire set of points on a simple 2-D graph, where nearby points in 1060-D space are still somehow near each other in 2-D. Remember that UMAP does not know they represent cars, or that they were 53 pixels across, or even that they were images at all. Here is a UMAP output chart from this data set, after about 20 seconds on a very ordinary laptop:



Each colored dot in the chart above represents one image of a car, and the local density of the dot groups says something about similarities in the data. Looking at how points cluster together, you can see that there are at least four main groups. Two smaller purple ones at left, and two larger green ones at right. It turns out, if you examine the original images corresponding to each point, these groups or clusters match obvious properties of the photos.

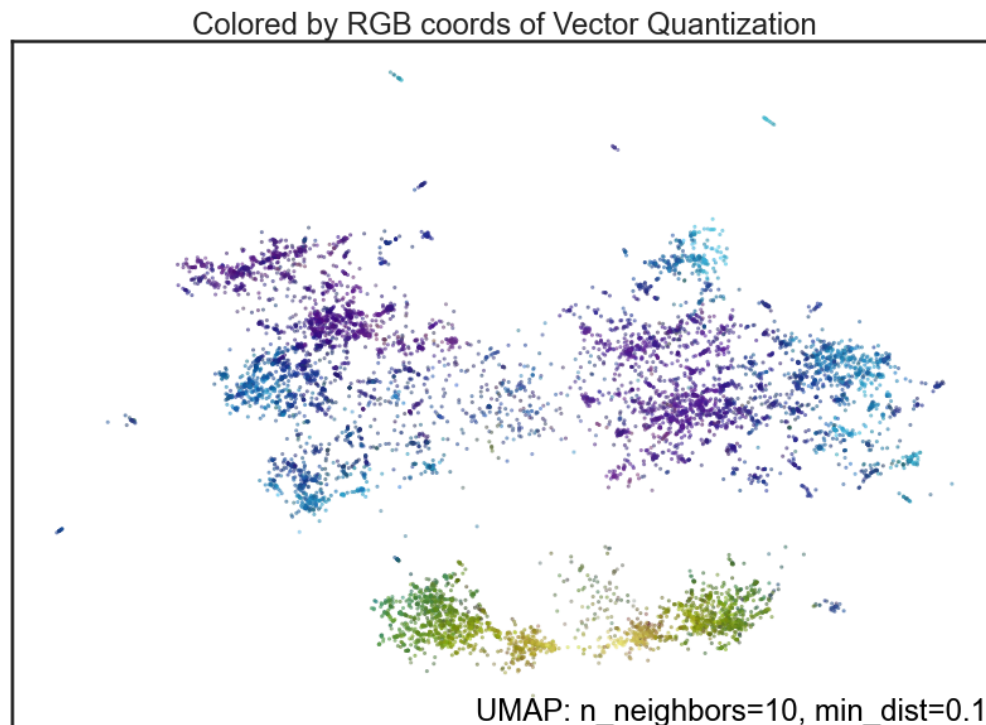
You can see this, if you lay a 10x10 grid of points over the image above, and look up the original photo corresponding to the closest mapped data point to each of those grid points. (A few at the edges are duplicates, where the data set is sparse.)



You can see that all the cars on the top half are facing to the left, and those on the bottom are facing to the right, with a middle row of different angles or partial images, where direction is less clear. Also, the photographs towards the left of the chart (matching the two purple blobs) were

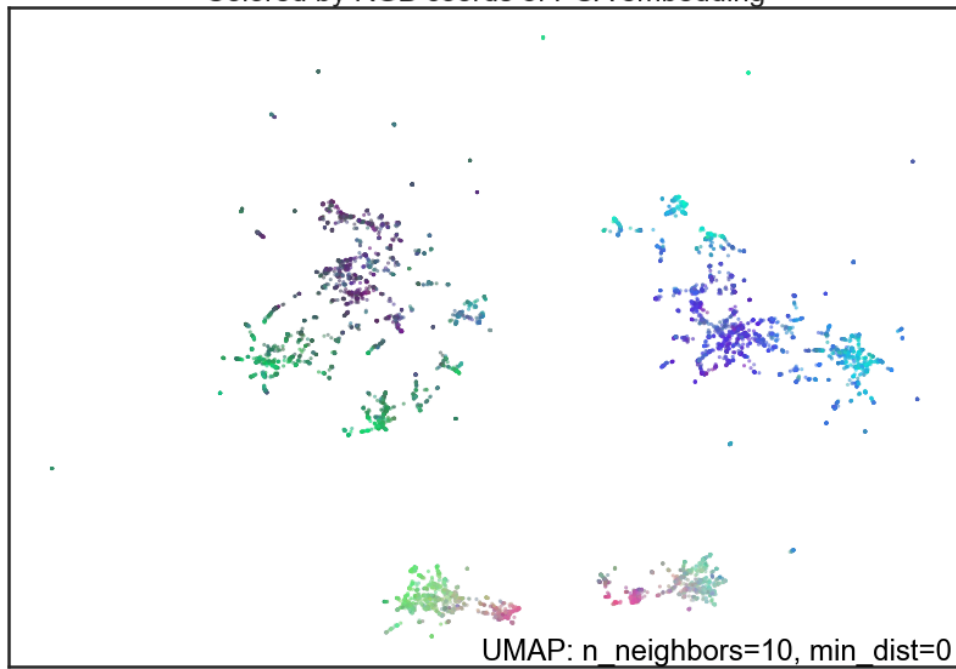
taken at night where fewer details are visible. Remember, unlike the neural-net type of AI computer vision that is trained with extensive compute time on large data sets to recognize cars or faces, UMAP has no training at all. It has no prior or built-in knowledge of cars or image processing generally.

The vertical and horizontal axes of the UMAP chart do not have a fixed meaning, and if you run the program again on the same data, you do not get the same output. The cluster shapes have similar features overall, but with many differences at smaller scales. Here is a second run, using a different random number seed. The plot looks mostly similar, but rotated 90 degrees.



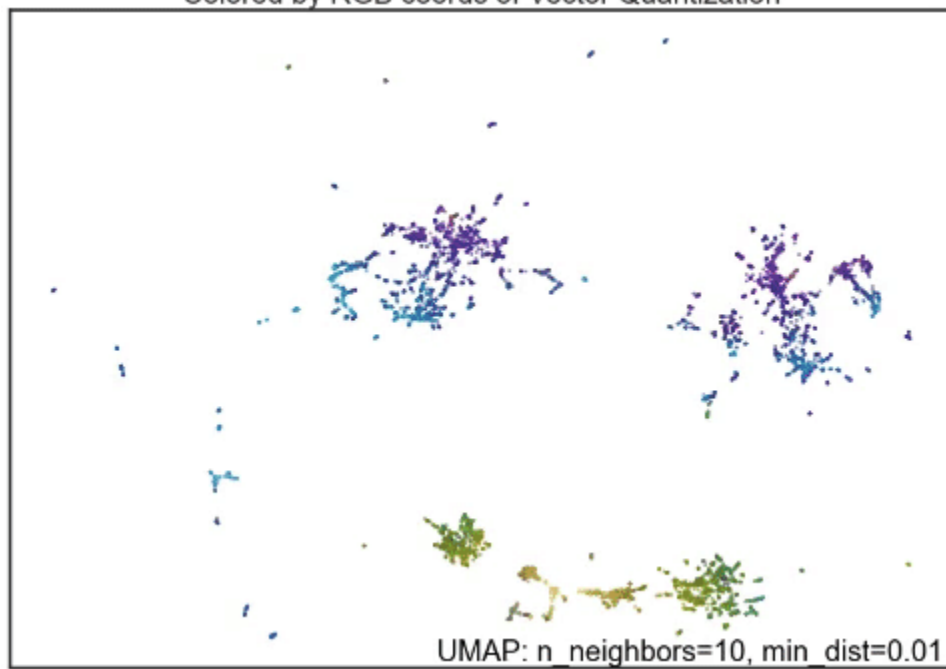
UMAP has some tuning parameters like "nearest_neighbors", "minimum_distance", and "density_lambda" which can be adjusted to separate out larger or smaller clumps of similarity within the data. The "neighbors" value is important for effective sorting of the data. Below is a chart using a different set of parameters but the same 10k data points. In this case, more of them are hidden because they are on top of each other. Nearby and overlapping points represent images that are very similar to each other, for example the same model of car but a different color, or the same car photographed at different times.

Colored by RGB coords of PCA embedding

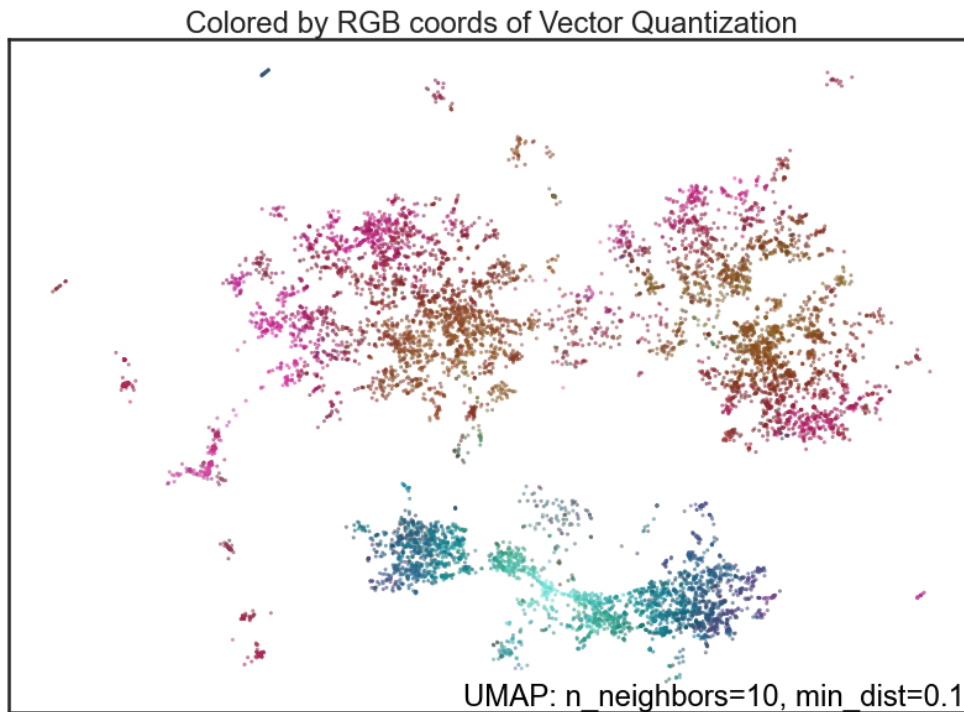


Below is an animation that shows how the output plot of the same data set changes with the "density_lambda" parameter ranging from 0.01 (denser) to 5 (more spread out).

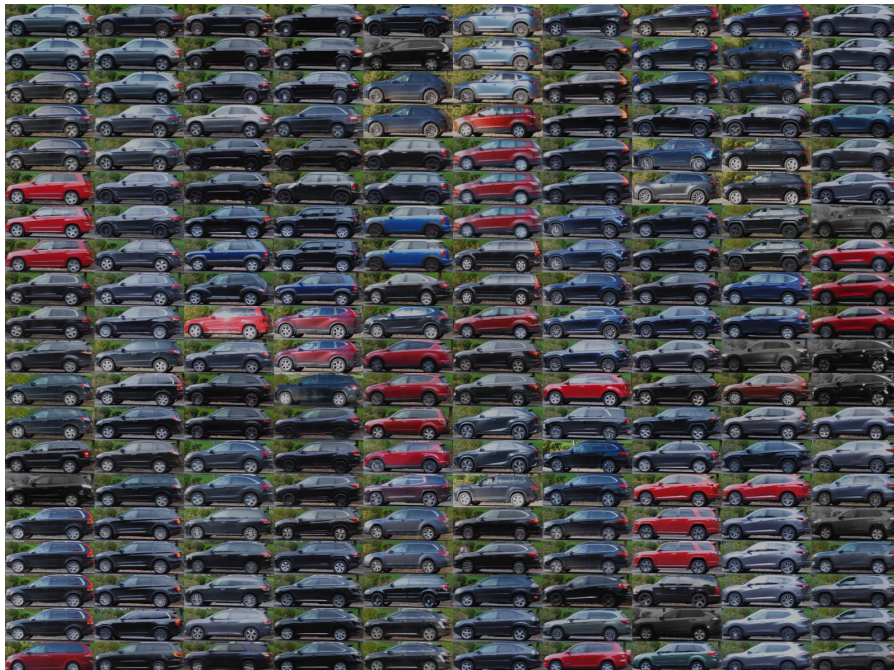
Colored by RGB coords of Vector Quantization



The plot below uses the first parameters, but a different set of 10k images taken weeks later from the same road. The same overall structure appears, and similar areas contain similar features. The isolated islands along the left are many nearly identical images of cars parked for periods of time, and partly obscured by a tree or a hedge.



If we zoom into one of the denser areas on the upper right hand blob for example, we might imagine seeing cars the same or very similar in style and tone next to each other, and extracting the corresponding original images for those points, in fact we do. Note in this experiment, UMAP had only the light/dark information shown in the first image set, and nothing about color.

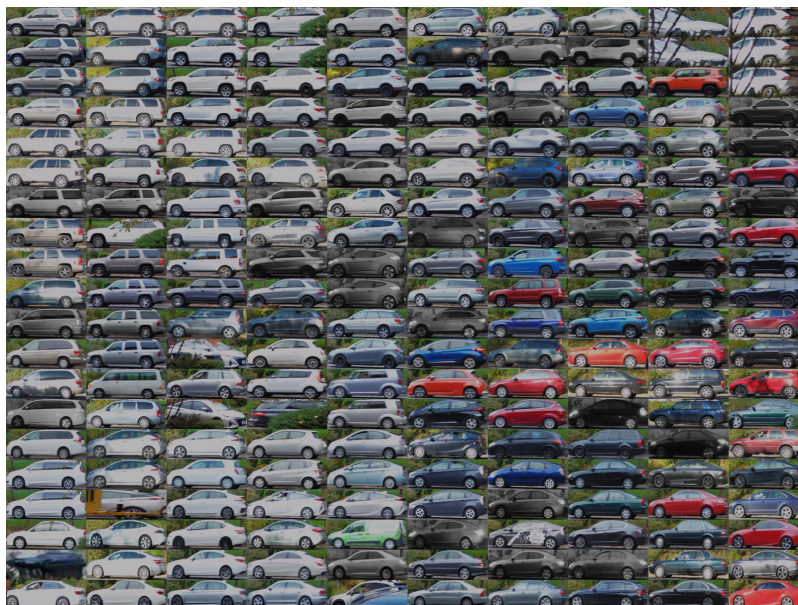


If we zoom into another nearby sub-blob, we see a selection of lighter-tone vehicles:



On each of the UMAP charts above, there are some small satellite blobs far from the main group. Those match images of cars at odd angles or partially blocked by trees. Cars nearer the main blob but on the fringes of it have an unusual, distinctive body shape, like a boxy Scion xB.

Here is another look at a larger area, enclosing most of the different kinds of the cars going left, and containing some others as well. At upper left are lighter-toned cars, and lower-right are darker. Cars close together here are in some sense similar to each other.

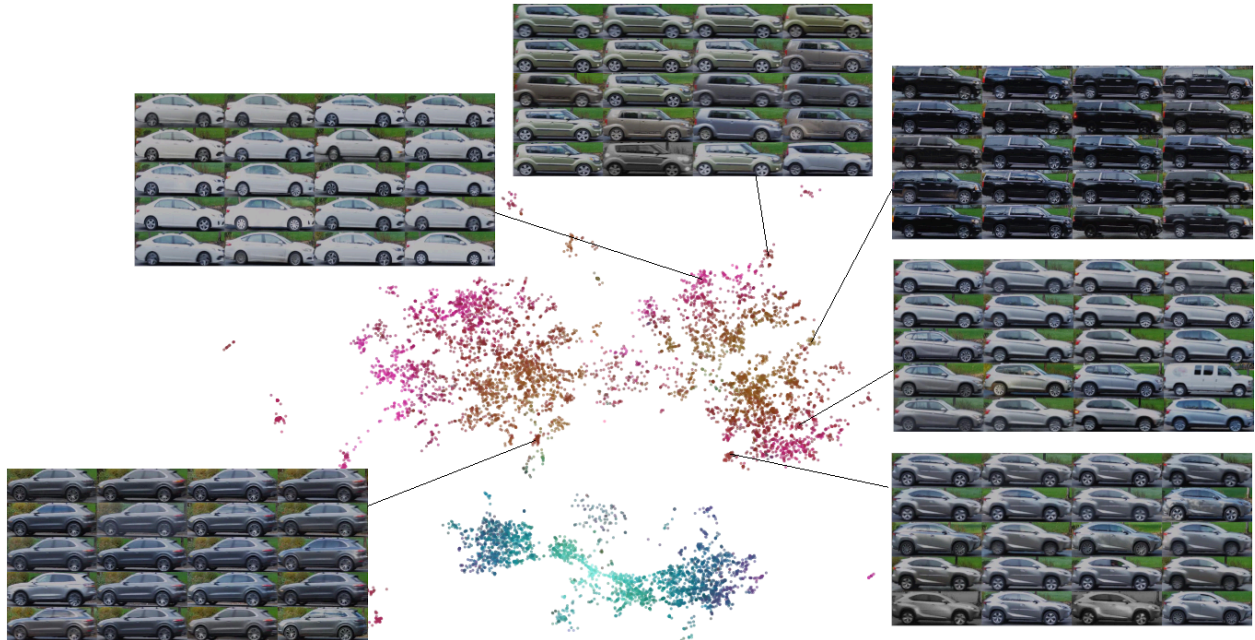


The results shown here are for output to a 2-D map, but you can select output into a 3-D space, or any other number of dimensions. That is harder to visualize, but may work better for additional computer analysis and region segmentation.

If you have one image or data point of interest, and want to find others in your data set most similar to it, you can simply find the nearest points on the UMAP chart. For example, I randomly selected one white SUV and asked for the 20 images closest to that. Note each of these 20 images are in fact different. Some appear to be either the same exact car seen at different times, or different instances of the same model.



You can do the same thing to find similar images using any chosen spot on the output graph:



If there are not enough close examples to a given data point, you get a wider variety.

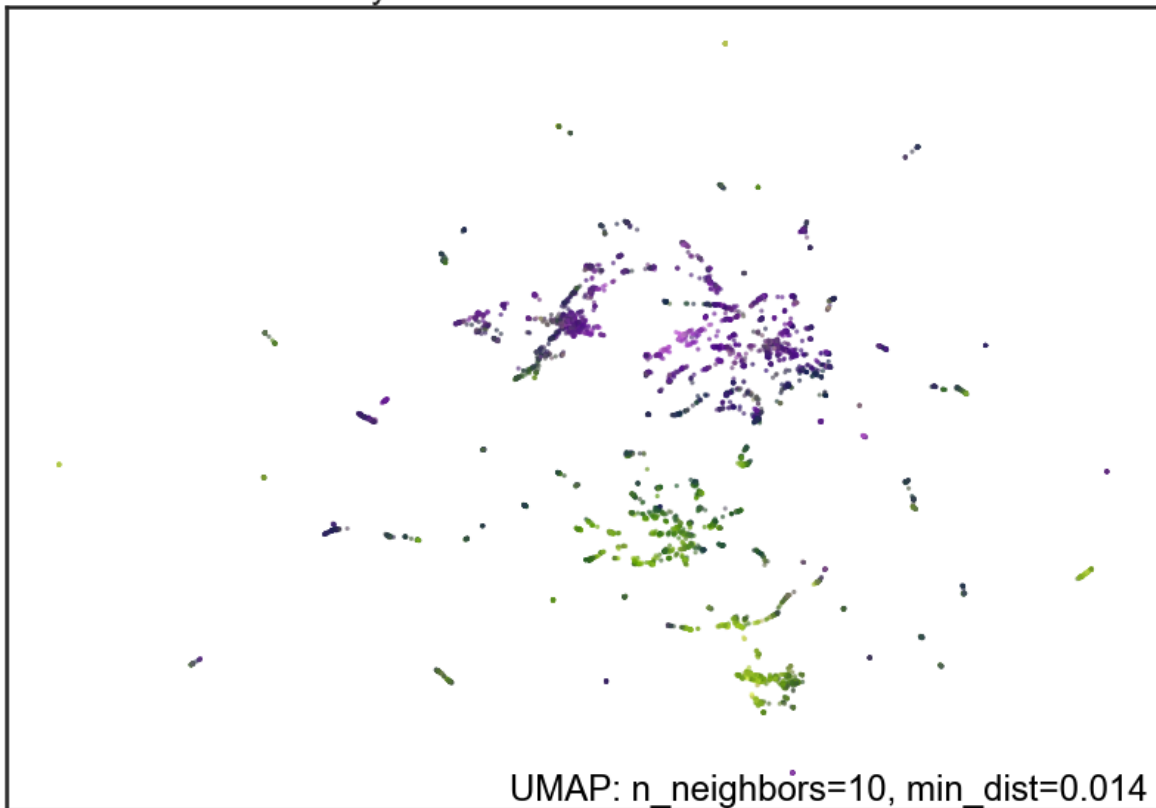


With category labels added, UMAP can also to some extent group the overall data set accordingly. Without them, it's still up to you to explore the geography of the map, to find and name "Scion Archipelago" or "Subaru Island" for example, but it's interesting how those places already exist in the data and UMAP can show them, without domain knowledge or supervision.



After first selecting out just daytime images going one direction, and re-running with `min_dist=0.014`, we get the map below. Each visible dot is generally 15 or more highly similar images, most of them apparently the same vehicle captured at different times.

Colored by RGB coords of Vector Quantization



There is an interactive museum exhibit online, which used UMAP in a somewhat similar way to group together historical images with similar features. You see the entire dataset and manually zoom in, or click on a specific category to move to that region in the collection.

<https://s3-us-west-2.amazonaws.com/lab-apps/pix-plot/index.html#MES26643>

Other examples are linked here: umap-learn.readthedocs.io/en/latest/interactive_viz.html

The UMAP program and some documentation: [umap-learn.
.io/en/latest/index.html](https://umap-learn.readthedocs.io/en/latest/index.html)