

Automatic Stylization Tool
User's Manual

Jacob Douglass, Matt Kolke, Xiaoran Wang, Zhouxing Liao

Installation Guide:

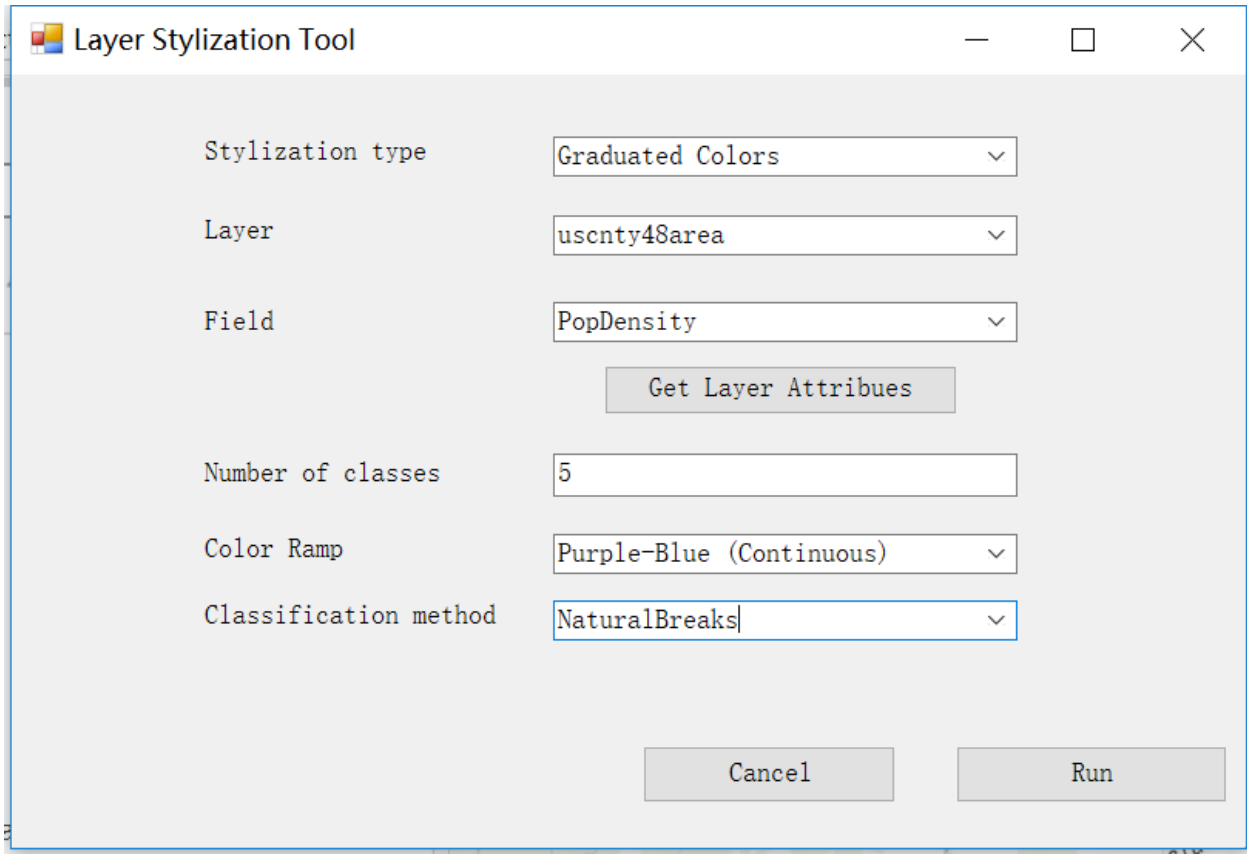
For proper usage ArcGIS Pro version 2.2 should be used. A version of Visual Studio that is from 2015 or later with ArcGIS Pro button add-ins rolled back to version 2.2 should be used if performing any alterations to the code itself.

Once the file is downloaded, navigate to the “bin” folder for the solution file and find the file “FinalProjectAddIn.esriAddInX”. Utilize a folder zip/unzip service to open the archive and find the “install” file. Doing this will save the add in to whatever well-known directory your machine has for ArcGIS Pro, you can double check by going to the well-known folder for ArcGIS Pro and checking to see if the add-in is present, if it is then it should be ready to go in ArcGIS Pro for use at any time.

User’s Guide:

The tool is compatible with any shapefile of polygon, point, and line layer data that can be loaded into ArcGIS Pro. For this tool to function properly the user must load the data into the map view first before trying to stylize it using the tool. Once the user has the data they wish to be stylized added to their project, the add in can be selected from the add ins tab at the main toolbar on the top of ArcGIS Pro. The user will then be presented multiple combo boxes to narrow down what type of data it is so as to stylize it best according to different style preferences that fit certain data types. The first drop list selects the stylization method and the second is to select which layer already in the map view is to be styled. The user then must click the “Get Layer Attributes” button to load all attributes of the selected layer into the “Field” list. This allows the user to choose the attribute they wish to be used when styling the map layer (note: if you decide to style a different layer before finishing the tool process, you must select “Get Layer Attributes” button again to load the correct attributes for the layer which has been selected). The last three boxes let the user pick the specifics as to how the layer will be styled. The first allows the user to enter how many classes they wish to be used, the second sets the specific color ramp desired, and the third allows for choices on classification methods for the data. A visual example of the interface is presented below.

User Interface



Note:

The user does not need to specify certain selections (or input) for certain style types, those required selections for each stylization type are as follows (checks indicate a required selection or input).

Stylization Types/Boxes	Layer	Field	Number of Classes*	Color Ramp**	Classification Method
Graduated Colors	√	√	√	√	√
Proportional Symbols	√	√			
Heat Map	√	√		√	
Unique Values	√	√		√	

* The number of classes should be an integer ranges from 1 to 32.

* For Graduated Colors Symbolization, color ramps with ‘continuous’ are preferred.

**For Heat Map, Color Ramps of ‘Yellow-Orange-Brown (Continuous)’ and ‘Yellow-Green-Blue (Continuous)’ are preferred. For Unique Value Symbolization, the ‘Set 3 (12 Classes)’ and ‘Paired (12 Classes)’ color ramps are recommended.

Discussion:

The automatic stylization tool is an add-in for ArcGIS Pro and it can be classified as a generic tool in ArcGIS Pro based on the characteristics on control, needs and update. It has the control of the development team; the development team can make any changes for the options in combo box; the periodical update would be applied to increase the utility and comprehension of the tool. As for the design, the development team directly designed the demo interface for the project, and the layout also followed the traditional design pattern of ArcGIS Pro tools.

The process of project development was done according to the agile development method. The team set up a general goal at the beginning without too many detailed plans in order to provide a higher degree of flexibility in the working process. At the later stages the team experienced refactoring and finally chose C#/ArcGIS Pro SDK as the most appropriate language and platform to achieve the goal. After finishing the basic requirements of the tool before the due date and having multiple rounds of testing, the team decided to add more levels to it, which included adding more map types and classification modes. During this process, the new versions were built and the new increments were added many times a day. These changes largely increased the functionality of the tool, thus, agile development helped maximize the working efficiency and project utility before the due date.

In terms of verification and validation, the development team conducted both software inspection and software testing. The author and owner also held the post of inspector and scribe roles to conduct the process of verification and validation. The original goal of the tool is to provide more logical and reasonable symbolization to the user, and then the software inspection proved that the tool fits for the intended purpose. For software testing, the development team chose system testing and applied three kinds of feature layers to the tool to check if it is performing as required across different data types. The team used polygon and point data to test all four stylizations and all classification modes, and then use polyline data to test the graduated colors. The tool performs when dealing with small scale data, such as the population of all

counties in Georgia and census tracts in Franklin County, Ohio. But as for the large-scale polygon data, like the data of all the United States counties, the number of color in unique value is relatively limited. In other words, when there are too many categories in the field, the number of color from *Set 3 (12 Classes)* and *Paired (12 Classes)* are not enough to support the number of categories, the categories that are not assigned colors would be automatically assigned to gray color. Also, the unique value performs better in dealing with polygon data than point data. The testing method is more likely to be the bottom-up integration testing since the team used lower-level functions to find out the issues of the final module. However in the end, according to the purposes of the project, the automatic stylization tool is able to fulfill the requirements and needs entirely.

Reference:

ArcGIS Pro SDK team (2019) ProSnippets Renderer. <https://github.com/esri/arcgis-pro-sdk/wiki/ProSnippets-Renderer>.