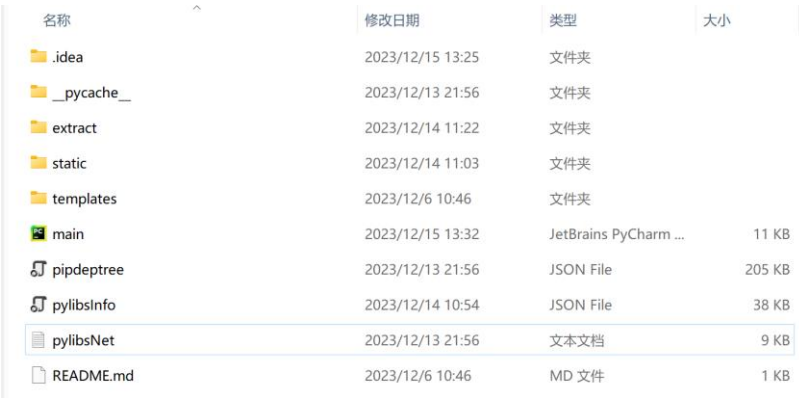
PyVisVue3D3: Python Visualization Using D3.JS and Vue3.JS

115305288@QQ.com https://pyvisvue3d3.yingshinet.com

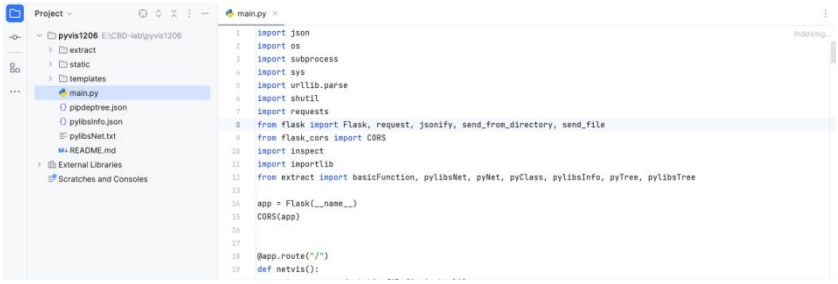
0- Download and Prepare for Running

When you download the PyVis, unzip it in your disk, such as:

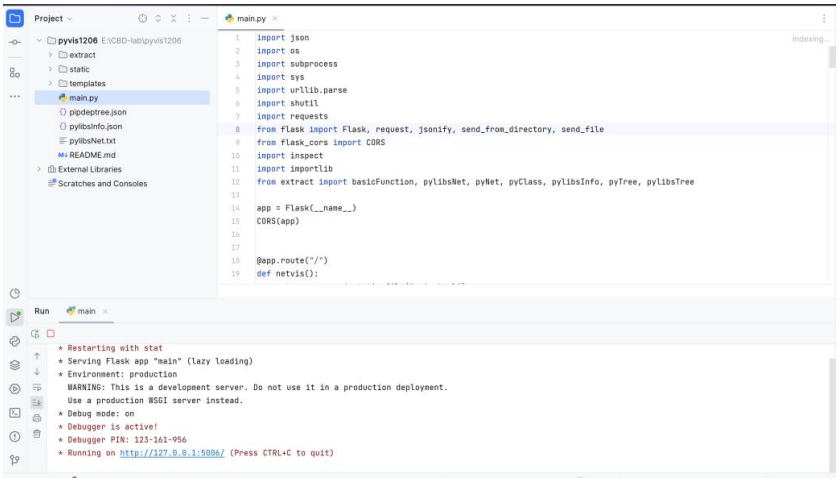


名称	修改日期	类型	大小
.idea	2023/12/15 13:25	文件夹	
__pycache__	2023/12/13 21:56	文件夹	
extract	2023/12/14 11:22	文件夹	
static	2023/12/14 11:03	文件夹	
templates	2023/12/6 10:46	文件夹	
main	2023/12/15 13:32	JetBrains PyCharm ...	11 KB
pipdeptree	2023/12/13 21:56	JSON File	205 KB
pylibsInfo	2023/12/14 10:54	JSON File	38 KB
pylibsNet	2023/12/13 21:56	文本文档	9 KB
README.md	2023/12/6 10:46	MD 文件	1 KB

Next please open the project in your own compile software, set pycharm as an example here.



Then install the python packages needed here: flask, flask-cors, requests, pipdeptree. Compile main.py to run the whole project. Other the third-party packages(such as PyTorch, torchvision, torchtext, transformers) are experiment data.



Open the link http://127.0.0.1:5006/ on the browser, let's explore the tool.

1- Extract Python package dependencies in your local environment

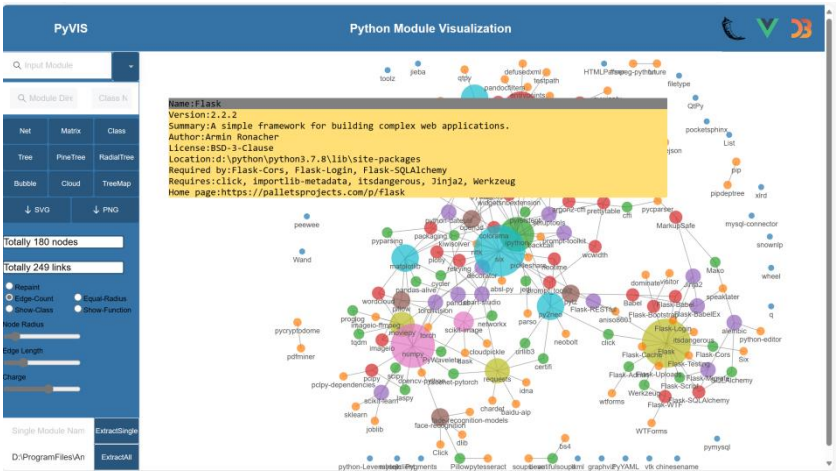
Focus on the left region and click the button [ExtractAll] to extract all packages in your local

environment, the backend programming will scan your site-packages directory and analyze all packages to generate some .json files provided for visualization. Also if you want to view details of certain package or to update the specific information, It's available to input the package name and click the button [ExtractSingle] to extract message of the package again.

2- Visualization for Python Virtual Environment(venv)

(1) Net for PylibsNet

Once open the link, the default whole environment packages visualization file would be called to show the relations between them with a force-directed graph. Each circle refers to certain package. Links refer to the relations between packages: requires or required by.

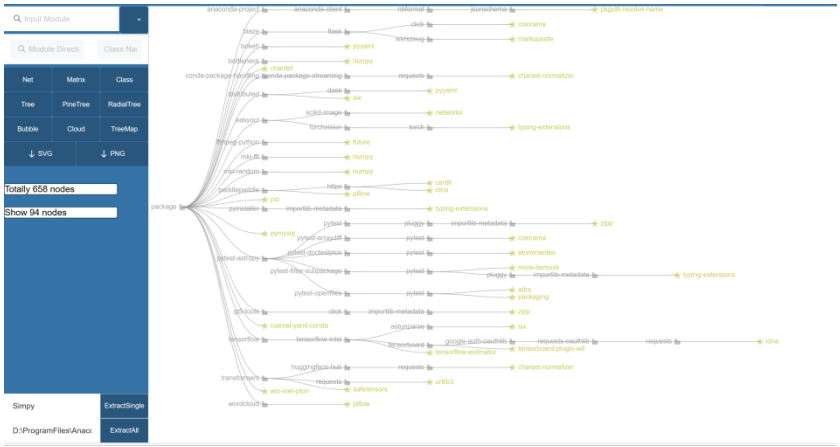


If you click the package name on the graph, the system will search the package selected automatically.

Also, the [Matrix] graph also shows the net json in adjacent matrix.

(2) Tree for PylibsTree

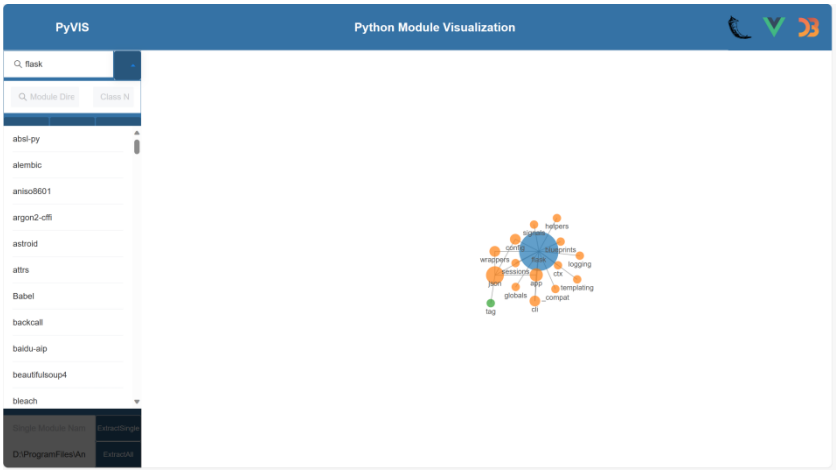
Click the [tree] button to show the hierarchical tree of the extracted tree json file(depending on pipdeptree). Different layouts are provides: PineTree, RadinalTree, Bubble, Cloud and TreeMap. The detailed message of each package will be described in Section 3.



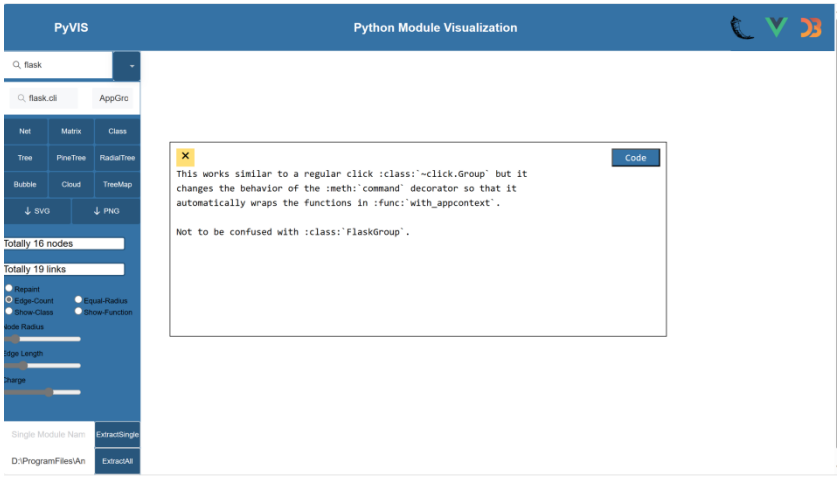
3- Single Package Visualization

(1) Generate the PyNet, PyTree, and PyClass in a package

Input the package name in the first search box and press enter, also it is available with dropdownlist. After that, the visualization on the package will be displayed.

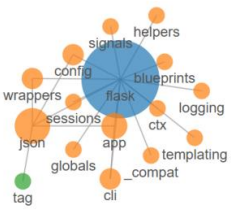


For the second search box, input the package directory in the left column, the class name in the right column(not necessary), thus it is available to retrieval python document and code at any directory and file.

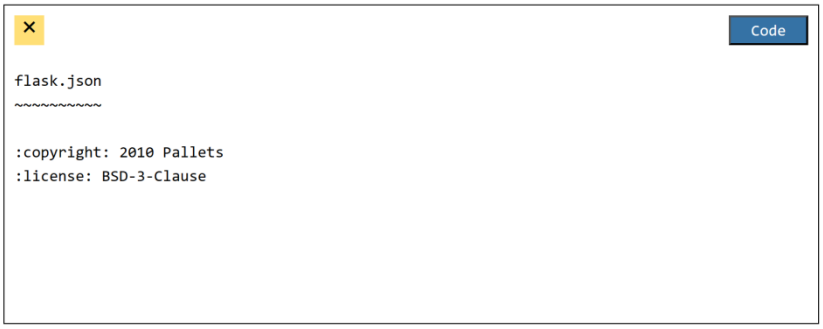


(2) Net: force-directed graph

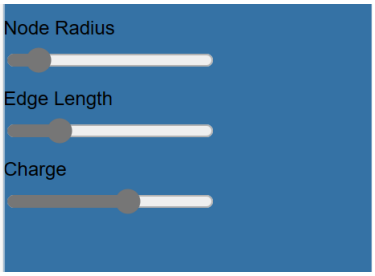
Click the [Net] button to display a force-directed graph illustrating the relationships among package. A pop-up window will show package information upon mouseover event. Additionally, a pop-up window will display the code and docs upon mouse down event.




```
Name:flask.json
Location:D:\ProgramFiles\Anaconda3\envs\newtorch\lib\site-packages\flask\json\__init__.py
Layer:1
Function:_dump_arg_defaults,_load_arg_defaults,_wrap_reader_for_text,_wrap_writer_for_text,detect
Class:JSONDecoder,JSONEncoder
```

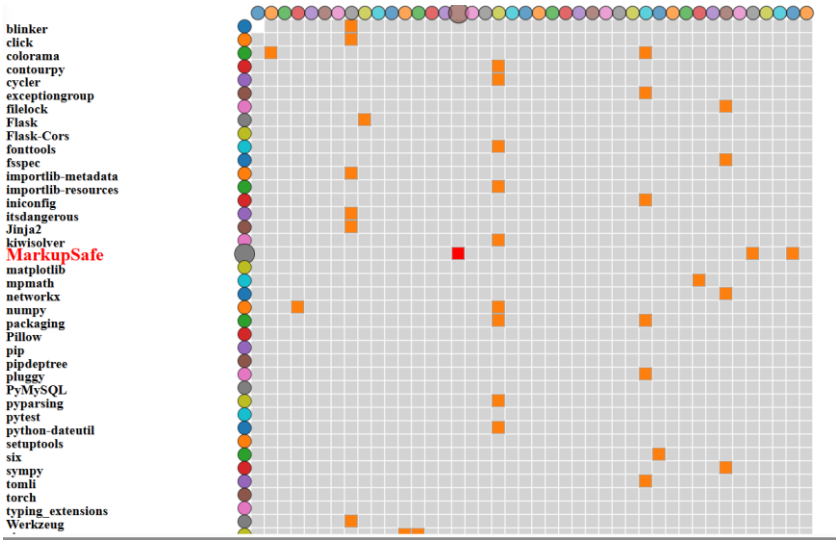


If you want to change the style of the force-directed graph, a control area list three option in which you can change the node radius, edge length and charge of the graph for better visual sight.



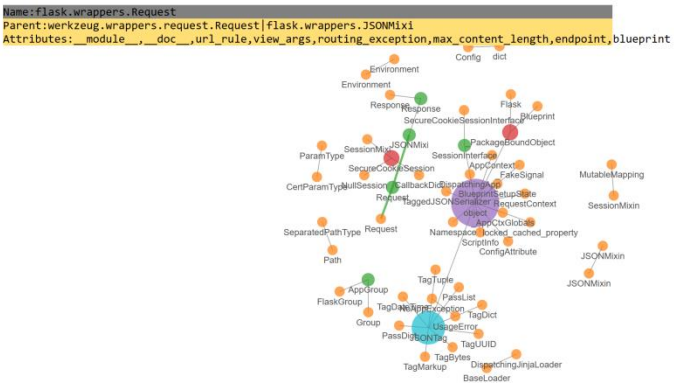
(3) Matrix visualization

Click the [Matrix] button to display a Matrix diagram illustrating the dependency relationships. The left area shows the names of packages and files. A pop-up window will show module information when the text is clicked. Additionally, The matrix points and circles have interactions with the text.



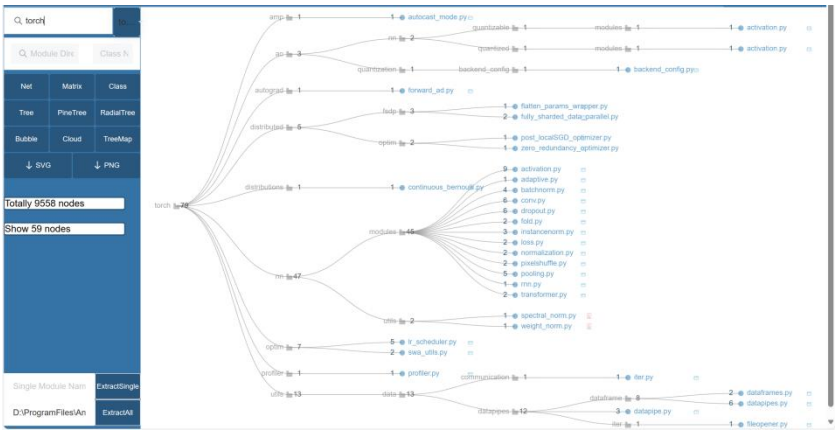
(4) Class force-directed graph

The class provide a force-directed graph that displays the information of all classes contained in the package, also you can call the information window by mouse-over option or call the document and code window. Control region is available as well as the [Net] module.

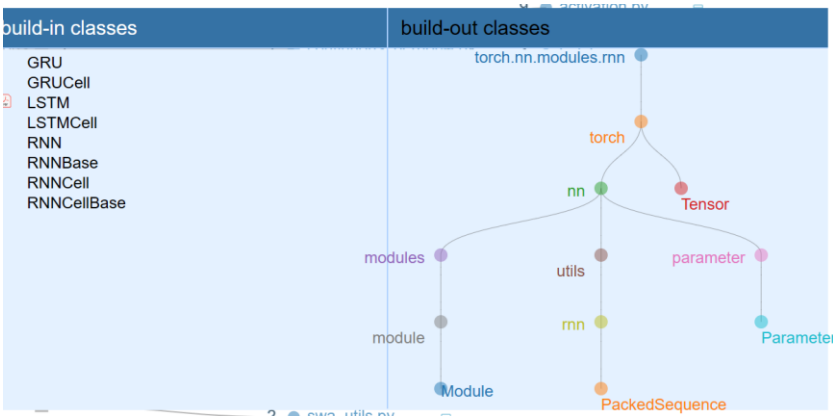


(5) Tree vis

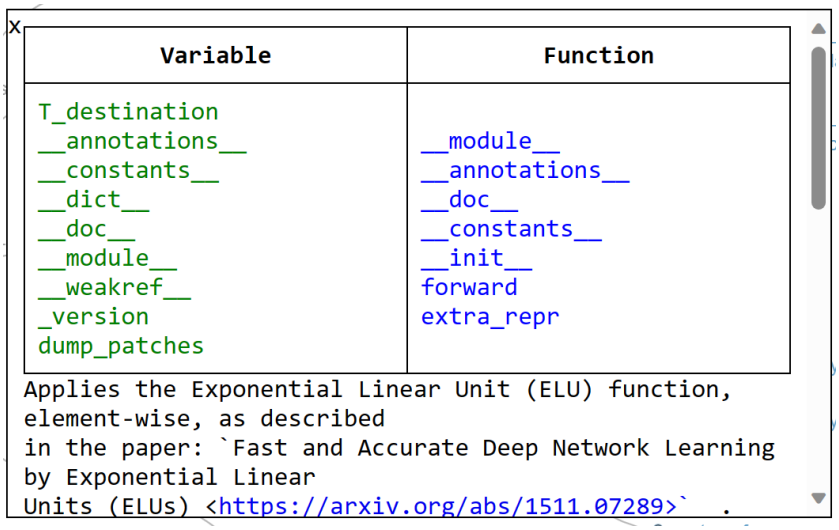
The tree displays a package folder's hierarchy, using the torch module as an example. The visualization is based on the extracted .json file. Each node is clickable; leaf nodes provide file links, while non-leaf nodes are extendable for easy visualization of large packages.



Clicking a leaf node opens a pop-up window with detailed information, including built-in and built-out class trees. Each class is clickable and leads to its code or documentation.



On click the build-in class name, a pop-up window which lists the variables, functions and document of the certain class name.

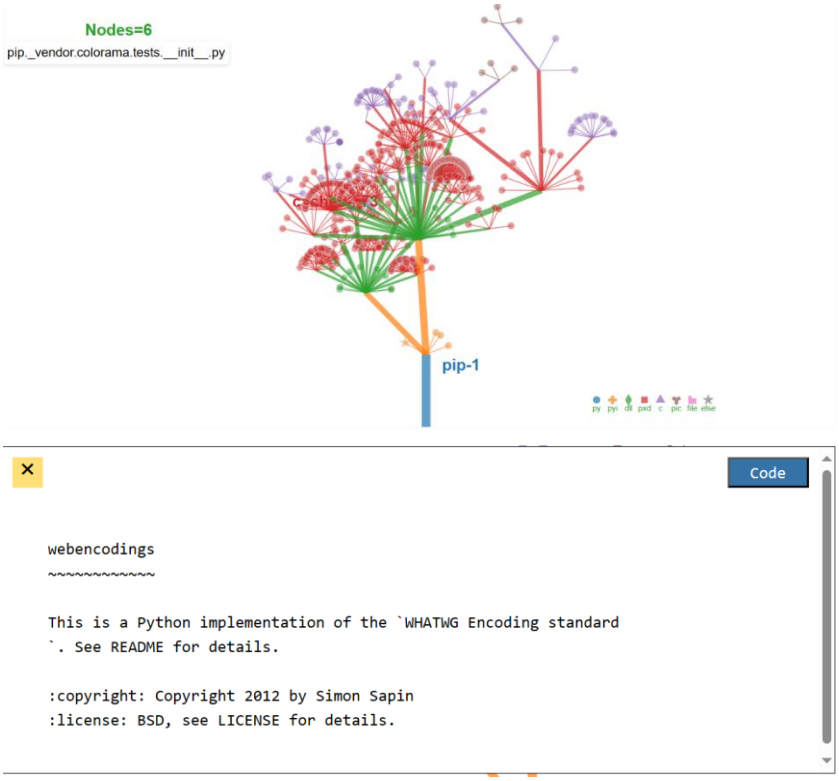


Variable	Function
T_destination __annotations__ __constants__ __dict__ __doc__ __module__ __weakref__ _version dump_patches	 __module__ __annotations__ __doc__ __constants__ __init__ forward extra_repr

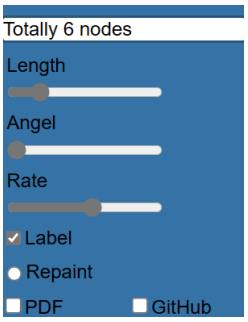
Applies the Exponential Linear Unit (ELU) function, element-wise, as described in the paper: `Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs) <https://arxiv.org/abs/1511.07289>` .

(6) PineTree vis

Clicking the [PineTree] button reveals a tree map of the directory and file hierarchy. Hovering over the module in the top left corner displays its full level information.

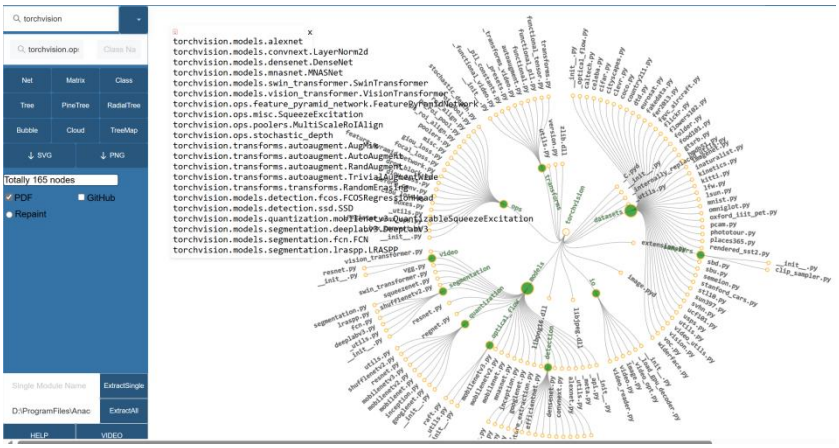


The left control area shows the number of nodes on the page , “Length”,“Angel”,“Rate”three adjustable sliders and the PDF and GitHub selection boxes appear.



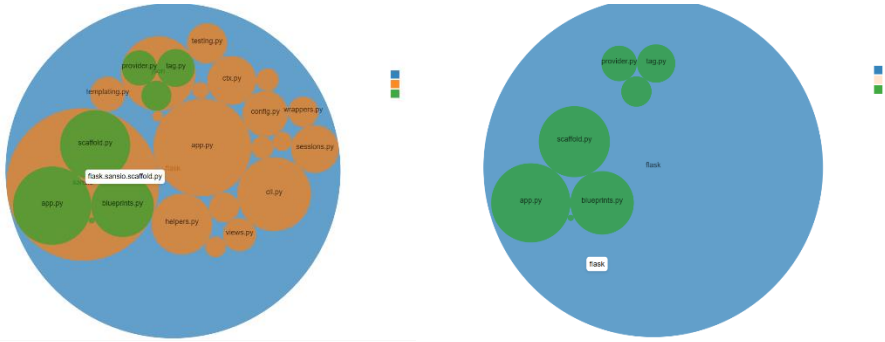
(7) RadialTree Module: radialtree visualization

Click the [RadialTree] button to show a radial tree layout, allowing interaction with nodes that have children.

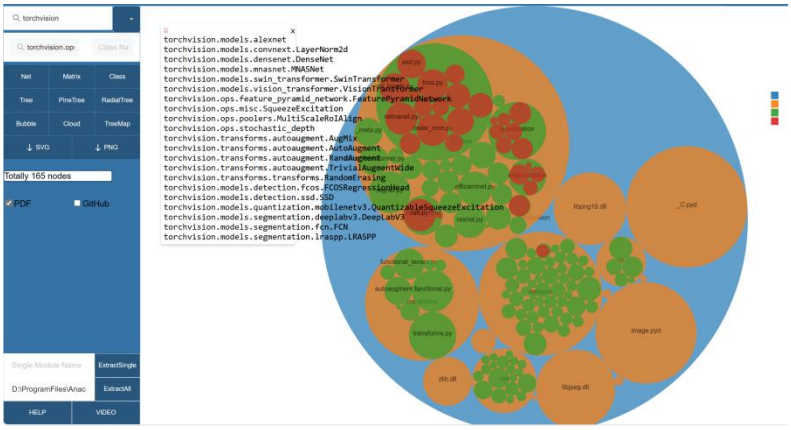


(8) Bubble Tree Vis

Click the [Bubble] button to view a Bubble diagram. Hovering over a module reveals its full level information, while clicking displays its code and documentation. The interactive colored buttons on the right allow layer switching.

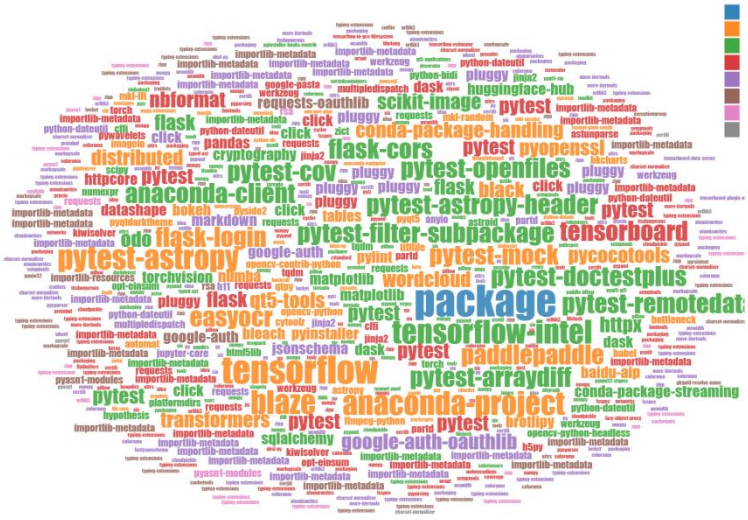


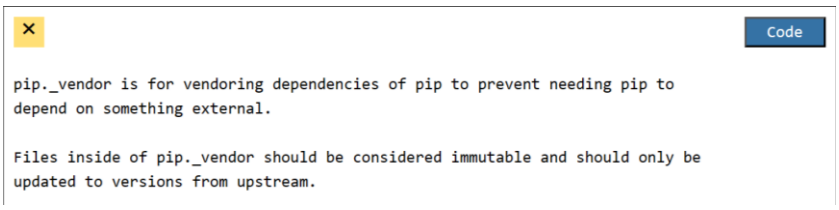
PDF file or GitHub file contained in the current package can be browsed.



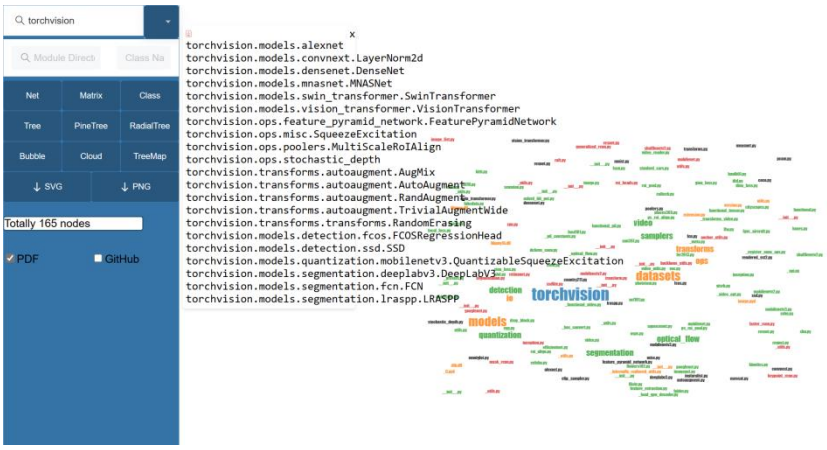
(9) Cloud: Wordcloud Vis

Click the [Cloud] button to display a word cloud map with interaction in color legend.





The PDF or GitHub can be shown.



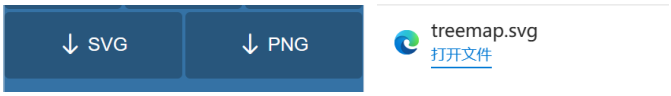
(10) TreeMap Vis

Click the [TreeMap] button to display a directory and file hierarchy. Bigger blocks represent larger values. Hover over a block to see its name and value, click to view its code and documentation. Use the left side zoom scale to adjust block sizes, and the Layer bar to change node colors. The number beside the Layer bar indicates the module's hierarchy. The Exchange button changes block colors, while the Raw button switches between raw and square root-processed data.

The left control area displays the number of nodes and options for PDF and GitHub. If selected, a pop-up window shows the PDF or GitHub links. If not, a prompt appears.

4- Download figures

We provide the download operation for each layouts.



Thanks for your attention and suggestion.

If you have any questions, feel free to contact us via email: 115305288@QQ.com.

2023.12.16