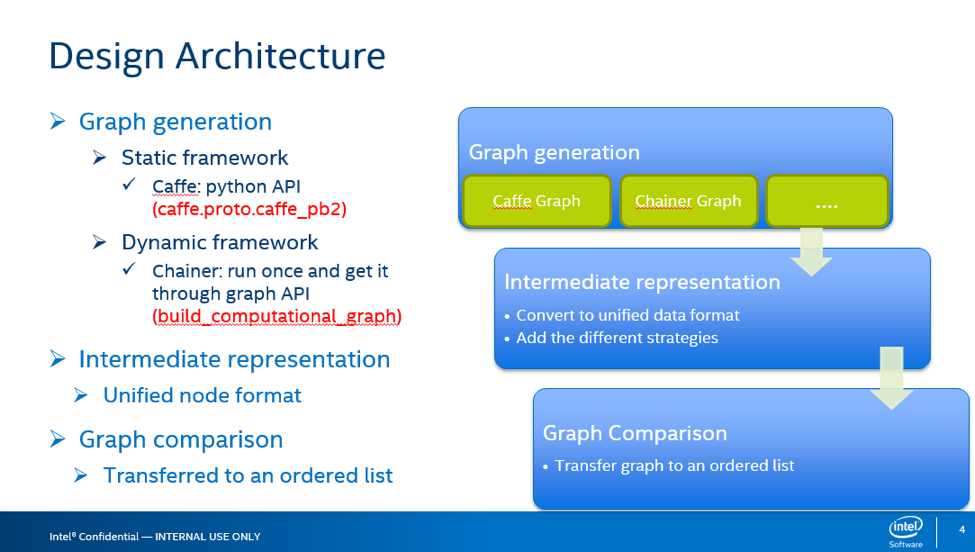
### Model Comparison Tool

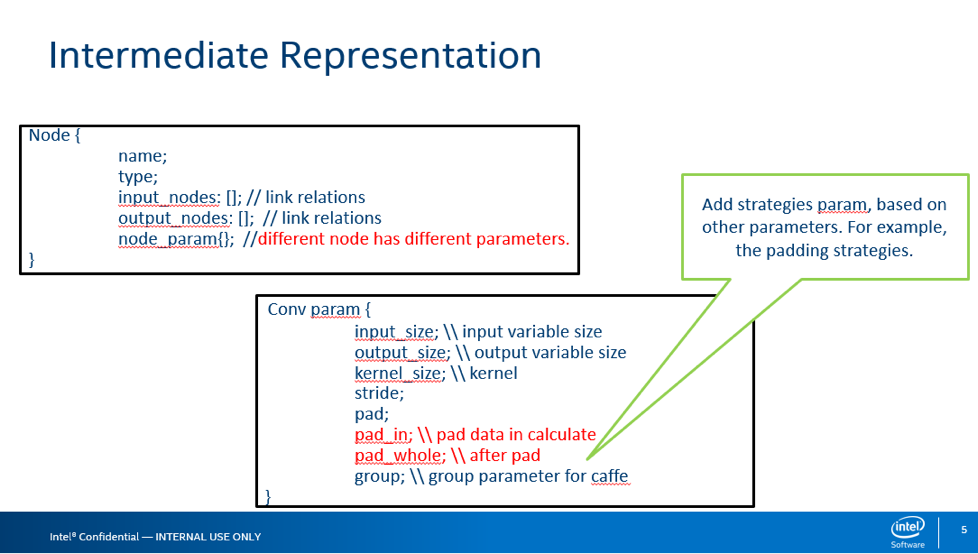
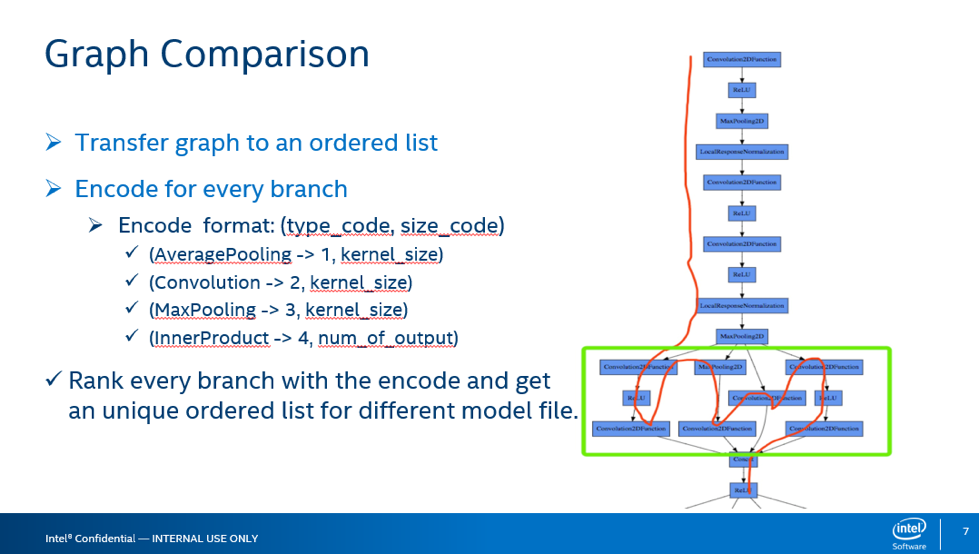
Model comparisontool is a tool to help developer to compare difference between two model from different framework. The tool support **Caffe, TF, MXNet, Chainer,pytorch.** We mainly compare the variable node shape and functionnode type. We also compare the computation the graph and padding strategy.

We also develop a web model comparison tool for user. User just need upload their models file and see the visually graph comparison result. Below Section 1 is the design detail of the tool and Section 2 is the Web version Detail, final is the Deploy part.

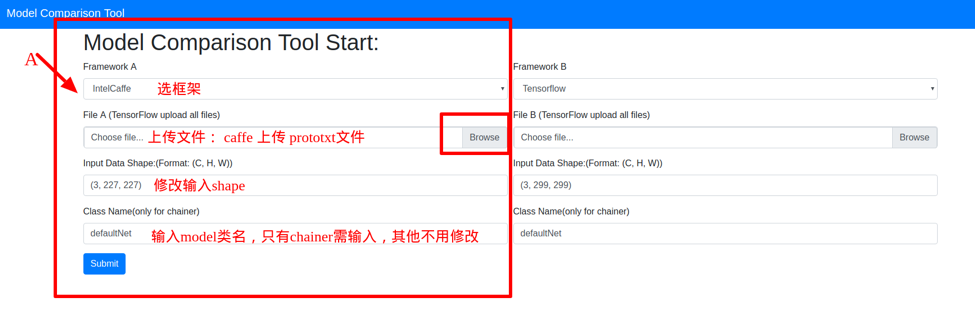
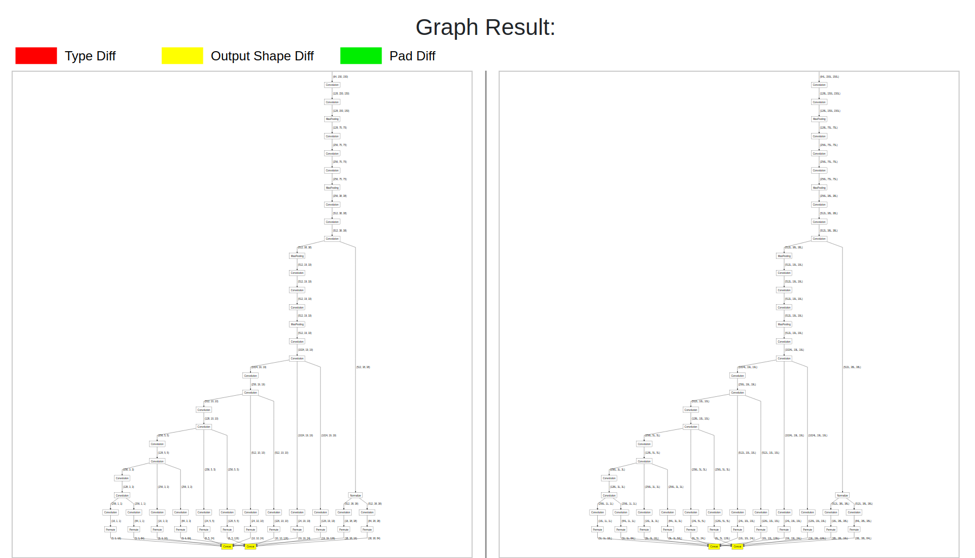
1. **Design Detail**
   * The whole tool based on the architecture as below.

* 
* dsdadasdasd
  + In our tool, firstlly loading the graph from different framework. You can find the method to load function responding to different framework as the Table below.

|  |  |  |  |
| --- | --- | --- | --- |
| * Framework | * Input Files | * Function file | * Detail |
| * Caffe | * train\_val.prototxt | * load\_caffe\_model.py | * Based on the caffe.proto.caffe\_pb2 to read model parameters. |
| * Tensorflow | * model save files | * load\_tf\_model.py | * Read computation graph from stored model file. So User should save the graph before. |
| * Chainer | * model class .py files | * load\_chainer\_model.py | * Get all functionNode from output. |
| * MXNet | * model json file | * load\_mxnet\_model.py | * Read json file and parameters. |
| * SSD | * SSD model | * load\_ssd\_\*.py | * SSD have special layer. |

* + Convert to unified data format and add different pdding strategies, as the Figure below. Later version, we add pw, ph, sh, sw, kh, kw params. source code in rank\_multi\_port.py.
* 
  + Graph comparison is a difficult problem. So we transfer graph to an ordered list and find all difference. Source code in rank\_multi\_port.py.
* 

1. Web Model Comparison Tool

* For a better User Experience, we develop a website. User just upload the input files and get the visual comparison result and detail parameters. The system combine **Django** backend and **Bootstrap** front end. Using **D3.js** to visualize the model graph. All source code in the *web-model-comparison-tool* folder.
* The website have two pages. In page 1, user upload their needing. Page 2 show the Graph result.
* 
* 

1. Deploy Method

* Our website verison deploy in developer01 server. The root username modeltools, password: abc110.
  1. ssh to the web server.
  + ssh modeltools@developer01
  1. cd to work folder.
  + cd ~/web-model-comparison-tool/model\_comparison\_tool/
  1. Start the uwsgi server.
  + uwsgi —socket /tmp2/model\_comparison\_tool.sock —module model\_comparison\_tool.wsgi —chmod-socket=777 —uid=www-data —gid=www-data
  1. Restart the nginx service.
  + sudo /etc/init.d/nginx restart
  1. Now you can browser the site: server local IP:8000 to use our tool.
* If wanna know more, please click on [HELP](https://uwsgi-docs.readthedocs.io/en/latest/tutorials/Django_and_nginx.html#configure-nginx-for-your-site).