## Splitting the Keras network

The Keras network should be split into smaller networks to allow debugging of the network at internal layers. The structure of these smaller networks should match the layer structure interpreted and run by the FPGA.

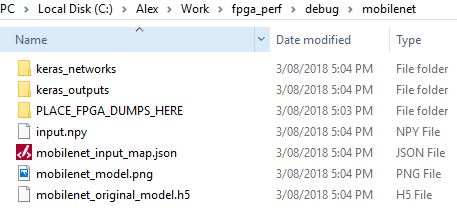


python tool/convertor.py tool/network/keras\_mobilenet.ini –debug 1 –input\_file image.jpg –r\_offs -127.5 –g\_offs -127.5 –b\_offs -127.5 –scale 0.0078431

Splitting a Keras network is performed by running the FPGA convertor tool with additional arguments.

* *debug:* Runs network splitting tool (set to 1)
* *input\_file:* An image that will be processed by the Keras network
* *r\_offs , b\_offs, g\_offs :* RGB offset (same values used for FPGA input preprocessing)
* *scale:* Image rescale factor (same value used for FPGA input preprocessing)

This splits the Keras network into multiple smaller networks to match the FPGA network structure, and runs the input\_file through the entire network. The output at each layer is saved as a numpy array.



This creates a debug folder for this network containing:

* Files:
  + *input.npy* - input image (saved as numpy array)
  + *input\_map.json* - structure of FPGA layers and inputs
  + *model*.*png*  - image of original Keras network structure
  + *original\_model*.*h5* - complete Keras model (identical to original network file)
* Folders:
  + *keras\_networks* - contains multiple Keras network, matching each layer of FPGA
  + *keras\_outputs* - contains outputs of each smaller Keras network
  + *FPGA\_DUMPS* - empty folder for FPGA dump files

## Get FPGA Data

Enable output dump on FPGA and run test image. Copy dump files to debug FPGA\_DUMP folder.

## Run FPGA Data each Keras layer

FPGA dumps can be run on each keras\_networks layer with tool/debug\_layer\_run.py and network folder input:



python tool/debug\_layer\_run/run\_debug\_layers.py debug/mobilenet

This creates a new folder in debug/network folder:

* *fpga\_dump\_debug\_outputs* – contains numpy outputs of each Keras mode

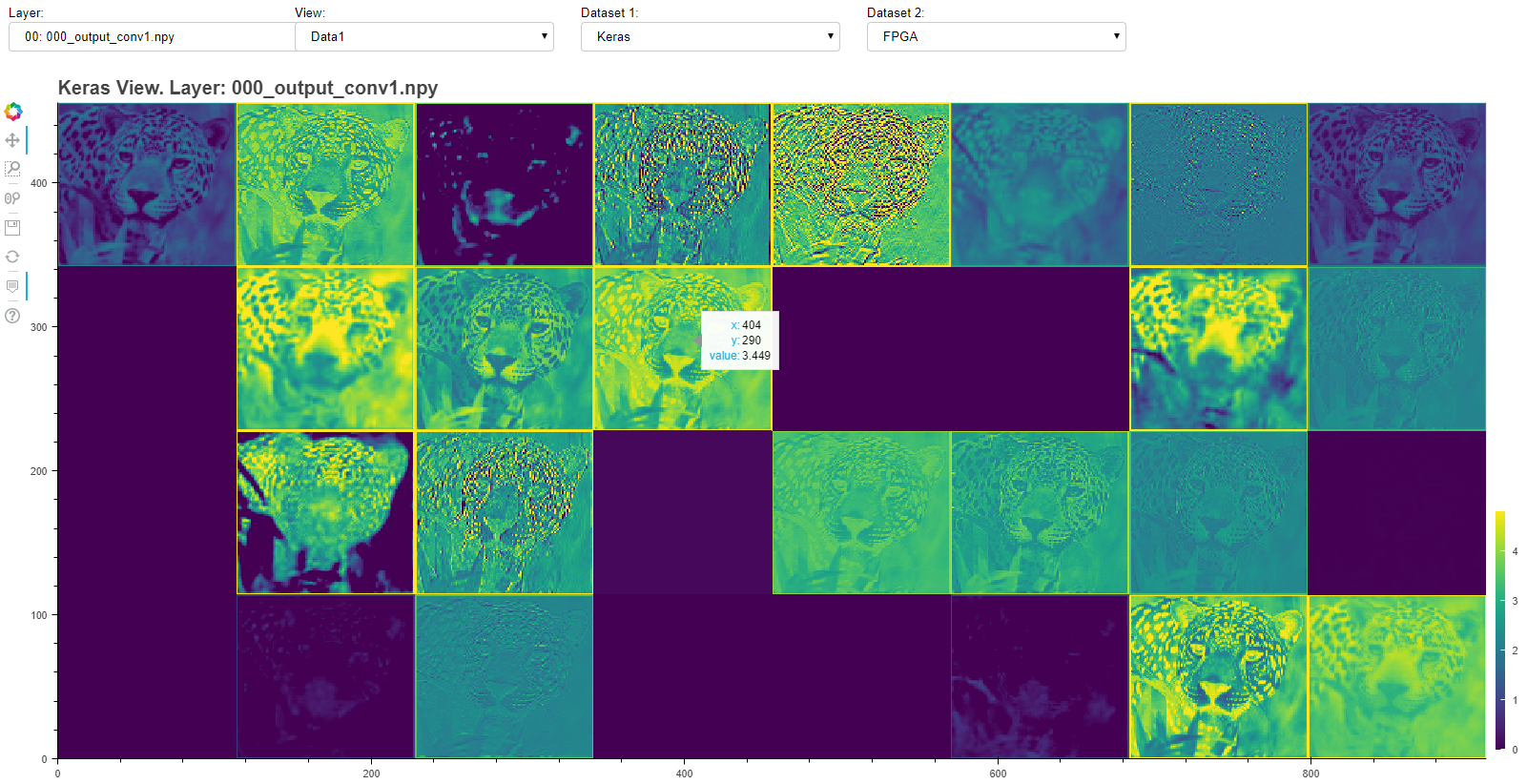
## Visualisation

Install Bokeh to set up the visualisation dependencies  
Run the interactive server using bokeh. Use the –show flag to launch the server, and provide the debug folder in -args



pip install bokeh

bokeh serve tool/layer\_check/visualisation –show –args debug/mobilenet



Available options are:

* Layer
* View
  + Data1/Data2 - Displays values of selected Dataset
  + Difference - Displays difference between selected datasets
  + Normalised difference - Displays a normalised difference from -1 to 1 using the following equation. Different activations are highlighted