```
#!/usr/bin/env python
""" vsmed Compute local 1x1x1 median using the buffer method """
import sys
from numpy  import *
from v4 import vx
from vxbuffer import *

of=' '
vxif=' '
clist = vx.vxparse(sys.argv,  "if= of= -v  - ")
exec (clist )

if 'OPT' in locals():
   print ("vsmed 1x1xn local temporal median filter")
   print ("if= input file")
   print ("of= output file")
   print ("[-v] verbose mode for very small images")
   exit(0)

optv = 'OPTv' in locals()

invx = vxIbuf(vxif, 3);
outvx = vxObuf( of );
im = invx.i
imr = empty( im[0].shape, dtype=im.dtype);
while invx.read():
    im = invx.i

    for t in range(im.shape[0] ):
        print(t)
        for y in range(im.shape[1]):
            for x in range(im.shape[2]):
                med = 0
                list_med = []
                for t in range (3):
                    list_med.append(im[t][y][x]) # append 3 data to list
                if max(list_med) == list_med[0]: # If index 0 is biggest
                    if list_med[1] > list_med[2]: # index 1 is second highest
                        imr[y][x] = list_med[1]
                    else: # index 2 is second highest or (1 and 2) is equal
                        imr[y][x] = list_med[2]
                elif max(list_med) == list_med[1]: # If index 1 is biggest
                    if list_med[0] > list_med[2]: # index 0 is second highest
                        imr[y][x] = list_med[0]
                    else: # index 2 is second highest or (0 and 2) is equal
                        imr[y][x] = list_med[2]
                else: # If index 2 is biggest
                    if list_med[0] > list_med[1]: # index 0 is second highest
                        imr[y][x] = list_med[0]
                    else: # index 1 is second highest or (0 and 1) is equal
                        imr[y][x] = list_med[1]


```

```
54          if optv:
55             print (imr)
56          outvx.add(imr)
57
58      outvx.close()
59
```