```
function atrous_convolution
(){
  /*Works by increasing the
 size of the filter by appe
nding zeroes to fill the ga
p between the parameters*/
  [Atrous Convolution](http
s://nanonets.com/blog/conte
nt/images/2020/08/main-qimg
-d9025e88d7d792e26f4040b767
b25819.png);
  Increase size by Dilation
 Rate;
  block{
    Dilation Rate = 1;
    3 x 3 Filter;
  }
  block{

    Dilation Rate = 2;
    5 x 5 Filter;

  }
  block{

    Dilation Rate = 3;
    7 x 7 Filter;

  }

  /*Last pooling layer has
stride = 1 instead of 2*/
  Downsample 8x only;
  Series of Atrous layers a
pplied to capture larger co
ntext;

  /*Downsampling by 8x to c
ompare each pixel*/
  Training;

  /*Use Bilinear upsampling
 to produce result of same
size*/
  Upsampling;
  Result Feature Map;
}

function spatial_pyramidal_
pooling(){
  /*Before SPP input images
 of multiple resolutions ar
e supplied to FCN. Feature
maps are then joined togeth
er to extract multi-scale i
nformation (Expensive compu
tation and time)*/



  Problem;

  /*Capture multiscale info
rmation from feature map us
ing single input image*/
  [Spatial Pyramidal Poolin
g](https://nanonets.com/blo
g/content/images/2020/08/sp
pnet.jpg);
  /*Input image of any size
 can be provided*/
  Advantage;

  /Single Input Image/;
  CNN Layers;
  Feature Maps;
  Spatial Pyramidal Pool;
  branch(a1){
    1x1;

  }
  branch(b1){
    2x2;
  }
  branch(c1){
    4x4;
  }
  join(a1,b1,c1);
  /*Concatenated by convert
ing to 1D vector, Gets mult
i-scale information*/
  concatenation;
}

function atrous_spp(){
  [Atrous Spatial Pyramidal
 Pooling](https://nanonets.
com/blog/content/images/202
0/08/deeplab_aspp.jpg);
  call spatial_pyramidal_po
oling();
  /*Fusing information from
 different scale and applyi
ng to atrous convolution. I
nputs are convolved with di
fferent dilation rates and
outputs are fused together*
/
  Concept;
  /Input Feature Map/;
  Convolve with;
  branch(a2){
    Filter Size (3x3);
    Dilation rate 6;
    Output1;
  }
  branch(b2){
    Filter Size (3x3);
    Dilation rate 12;
    Output2;
  }
  branch(c2){
    Filter Size (3x3);
    Dilation rate 18;
    Output3;
  }
  branch(d2){
    Filter Size (3x3);
    Dilation rate 24;
    Output4;
  }
  join(a2,b2,c2,d2);
  /*Outputs concatenated du
e to same size*/
  Concatenation;
  Fused Output;
  /*Stride to keep downsamp
le 8x*/
  Convolve with 1x1 map;
  Upsampling;
  /*To keep Global Contextu
al Information*/
  Output1;
  /*Get the required number
 of channels*/
  Pass through 1x1 Convolut
ion;
  Improved Results;

}

/*Multitude techniques for
results improvement by Goog
le*/
Deeplab Neural Network;
block{

branch(theory){
  /*Three main improvements
*/
  Theory;
  /*Problem:
  Excessive downsizing due
to consecutive pooling. (Do
wnsampled by 32x)*/
  Loss of Information;
  /*Upsampling 32x will cau
se*/
  Compute and Memory expens
ive operation ;

  /*Solution to understand
large context using same nu
mber of parameters*/
  [Atrous convolution](http
s://nanonets.com/blog/conte
nt/images/2020/08/main-qimg
-d9025e88d7d792e26f4040b767
b25819.png);
  call altrous_convolution(
);

  Altrous spatial pyramidal
 pooling;

  call spatial_pyriamidal_p
ooling;
  call atrous_spp();

  ||Conditional Random Fiel
d||;

}
}
block{
branch(logic)
{
  Logic;
}
}
join(theory, logic)
```