```
function [bn0,bn1,bn2,computed_ec] = bettinmbrs(img,n)
    ch = n;
    bn0 = 0; bn1 = 0; bn2 = 0; computed_ec = 0;

    switch(ch)
        case 1
            sampleMat1 = img;
            betti0 = 0; betti1 = 0; betti2 = 0;
            %sampleMat1 = abs(sampleMat1 - 1); %Grains White
            [l,betti0] = bwlabel(sampleMat1,8) ;

                r = regionprops(l);
                rr = struct2cell(r(:,1));
                rr = cell2mat(rr(1,:));
                co = 0;
                for x = 1:1:length(rr)
                if rr(x) == 1
                co = co + 1;
                end
                end
                betti2 = co;

            for c = 1:1:betti0
                clust = ismember(l,c);
                k = 1; rw = 0; col = 0;
                count = 0; smclust = 0; ii = 1; jj = 1;
                for i = 1:1:900
                    for j = 1:1:900
                        if clust(i,j) == 1
                            col(k) = j ;
                            k = k + 1;
                        end
                    end
                end

                for i = rw(1):1:rw(end)
                    for j = col(1):1:col
                        smclust(i-rw(1)+1,j-col(1)+1) = clust(i,j);
                    end
                end
                smclust;
                smclust = abs(smclust - 1);
                [sl,sbetti1] = bwlabel(smclust,8) ;
                betti1 = betti1 + sbetti1 ;
            end

              bn2 = betti2;
              bn0 = betti0;
              bn1 = betti1;
              computed_ec = betti0-betti1+betti2 ;

        case 2
            sampleMat1 = abs(img-1);
            betti0 = 0; betti1 = 0; betti2 = 0;
            [l,betti0] = bwlabel(sampleMat1,8) ;
                r = regionprops(l);
```

```
                rr = struct2cell(r(:,1));
                rr = cell2mat(rr(1,:));
                co = 0;
                for x = 1:1:length(rr)
                if rr(x) == 1
                co = co + 1;
                end
                end
                betti2 = co;

            for c = 1:1:betti0
                clust = ismember(l,c);
                k = 1; rw = 0; col = 0;
                count = 0; smclust = 0; ii = 1; jj = 1;
                for i = 1:1:900
                    for j = 1:1:900
                        if clust(i,j) == 1
                            rw(k) = i ;
                            col(k) = j ;
                            k = k + 1;
                        end
                    end
                end

                for i = rw(1):1:rw(end)
                    for j = col(1):1:col(end)
                        smclust(i-rw(1)+1,j-col(1)+1) = clust(i,j);
                    end
                end
                smclust;
                smclust = abs(smclust - 1);
                [sl,sbetti1] = bwlabel(smclust,8) ;
                betti1 = betti1 + sbetti1 ;
            end
              bn2 = betti2;
              bn0 = betti0;
              bn1 = betti1;
              computed_ec = betti0-betti1+betti2 ;
    end
end
```

```
function expected_ec = eulercharact(img,n)
    ch = n; expected_ec = 0;
    switch(ch)
        case 1
            expected_ec = bweuler(img,8);
        case 2
            img = abs(img - 1);
            expected_ec = bweuler(img,8);
    end
end
```

```
function fd = fractaldim(img,n)
     ch = n;
     fd = 0;
    switch(ch)
        case 1
            A = img; % Black->Pores-> 0  and  White->Grains-> 1
            leni = length(img);
            S = leni; %Total Size
            c = 1;
            %%%Calculating Factors%%%
            for n = 1:leni
                if rem(leni,n) == 0
                    factr(c) = n ;
                    c = c + 1;
                end
            end
            lenf = length(factr);
            %3D Box Counting Fractal Dimension Calculation
            for s = 1:1:lenf
                x(s) = factr(s); m = 0; %Scaling factor
                r = factr(s);

                for i = 1:S/r:S
                    for j = 1:S/r:S
                            B = A(i:i + S/r - 1, j:j + S/r - 1);
                            if sum(sum(B)) > 0
                                m = m+1;
                            end
                    end
                end
                N(s) = m; %Number of boxes count
            end
            p=polyfit(log(x(2:lenf)),log(N(2:lenf)),1); %Slope of line
            fd = p(1);

        case 2
        A = img; % Black->Pores-> 0  and  White->Grains-> 1
            leni = length(img);
```

```
            S = leni; %Total Size
            c = 1;

            %%%Calculating Factors%%%
            for n = 1:leni
                if rem(leni,n) == 0
                    factr(c) = n ;
                    c = c + 1;
                end
            end

            %3D Box Counting Fractal Dimension Calculation
            for s = 1:1:lenf
                x(s) = factr(s); m = 0; %Scaling factor
                r = factr(s);

                for i = 1:S/r:S
                    for j = 1:S/r:S
                            B = A(i:i + S/r - 1, j:j + S/r - 1);
                            if prod(prod(B)) == 0
                                m = m+1;
                            end
                    end
                end
                N(s) = m; %Number of boxes count
            end
            p=polyfit(log(x(2:lenf)),log(N(2:lenf)),1); %Slope of line
            fd = p(1);
    end
end
```