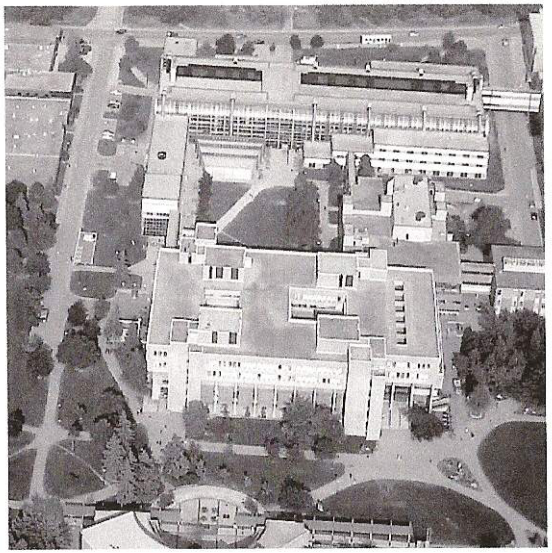
A3, Task 4


```
% Function compress
%
% Input
%    f   is the input image, a matrix of numbers
%    tol is the drop tolerance (bigger value = more compression)
%
% Output
%    g    is the output image
%    drop is the fraction of Fourier coefficients dropped
%         (i.e. with modulus less than tol)
%
function [g, drop] = compress(f, tol)

    [h,w] = size(f);  % returns the width and height of f
    g = zeros(h,w);

    drop = 0;

    for x = 1:16:w
        for y = 1:16:h
            f16 = f(y:(y+15),x:(x+15));

            F = fft2(f16); % Compute DFT of f16

              % Create a mask such that
              %   = 1 if that element in abs(F) is >= tol
              %   = 0 if that element in abs(F) is < tol
              mask = ( abs(F) >= tol );

              % Apply mask to F to zero out smallest Fourier coefs
              Fthresh = mask .* F;

              % Reconstruct compressed f16 using the inverse FFT
              g16 = ifft2(Fthresh);

            % Calculate how many coefficients were dropped
            % (Assume all Fourier coefs are nonzero)
            drop = drop + sum(sum(1 - mask));

            % Copy reconstructed tile to corresponding location in g
            g(y:(y+15),x:(x+15)) = (g16);
        end
    end

    % Calculate drop RATIO
    drop = drop / (w*h);
```

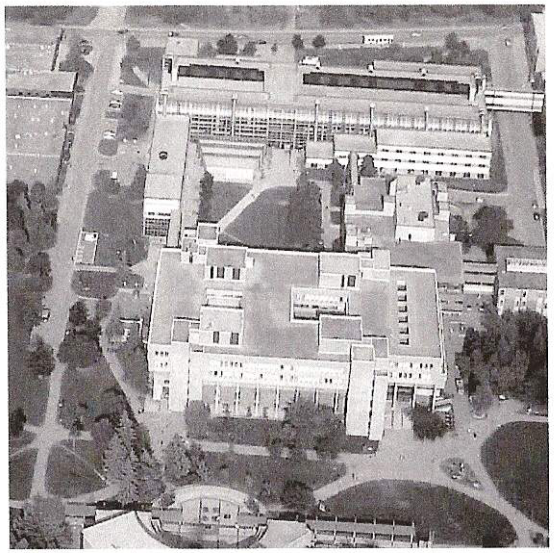
```
% CS 370 Assignment 3, Task 4
% compression_demo.m

f = double(imread('mc_dc.jpg'));

tol50 = 170;     % For drop = 50
[g50 drop50] = compress(f,tol50);
drop50

tol75 = 370;  % For drop = 75
[g75 drop75] = compress(f,tol75);
drop75

tol95 = 1150;     % For drop = 95
[g95 drop95] = compress(f,tol95);
drop95

tol99 = 3200;     % For drop = 99
[g99 drop99] = compress(f,tol99);
drop99

% Display them all
figure(1); orient landscape;

subplot(2,2,1); colormap(gray(256));
image(g50); axis image; axis off; title(['drop = ' num2str(drop50*100)
'%']);

subplot(2,2,2); colormap(gray(256));
image(g75); axis image; axis off; title(['drop = ' num2str(drop75*100)
'%']);

subplot(2,2,3); colormap(gray(256));
image(g95); axis image; axis off; title(['drop = ' num2str(drop95*100)
'%']);

subplot(2,2,4); colormap(gray(256));
image(g99); axis image; axis off; title(['drop = ' num2str(drop99*100)
'%']);
```

After trial and error to see what value of tol achieves a particular drop percentage.

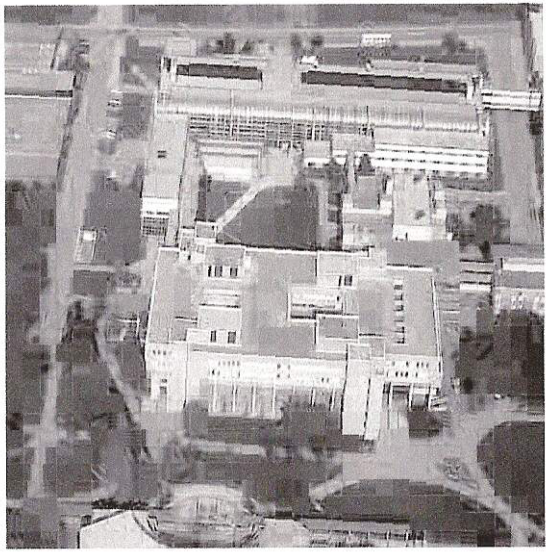
drop = 50.6756%



drop = 75.4467%



drop = 94.8963%



drop = 99.0311%