

1.

a)

```
function d = SSD(R1, R2)
if size(R1) == size(R2)
    d = sum(sum((R1 - R2).^2)) ;
else
    d = NaN ;
end
```

Result)

```
R1 = [1, 0, 0, 2; 2, 1, 0, 1]; R2 = [0, 2, 1, 1; 1, 0, 0, 1];
SSD(R1,R2)
ans =
    9
```

b)

```
function A = brute_force_ssd(T,I)

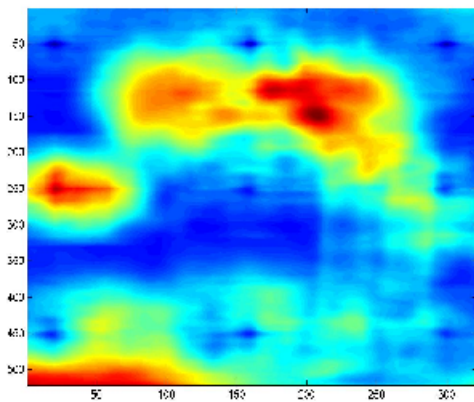
Range = size(I)- size(T) + 1 ;

A = zeros(Range) ;
size_template = size(T) ;

for i = 1 : Range(1)
    for j = 1 : Range(2)
        A(i,j) = SSD(I( i : i + size_template(1) - 1, j : j + size_template(2) - 1), T);
    end
end
```

Result)

```
CA=brute_force_ssd(double(imread('template.tif')), double(imread('Image.tif')));
Imagesc(CA);                                imshow(CA/max(max(CA)));
```



c)

```
function Cpt = find_best(SSD_value, Npt, Wsize)
```

```
[r c] = size(SSD_value);
```

```
tsize = r * c;
```

```
tDV = [];
```

```
for i = 1 : r
```

```
    tDV = [tDV SSD_value(i,:)];
```

```
end
```

```
[V I] = sort(tDV);
```

```
Fpt = 0;      Cpt = [];      i = 1;
```

```
while Fpt < Npt && i ~= tsize
```

```
    row = floor((I(i) - 1) / c) + 1;
```

```
    col = mod(I(i), c);
```

```
    if col == 0
```

```
        col = c;
```

```
    end
```

```
    valid = true;
```

```
    for j = 1 : Fpt
```

```
        valid = ( row <= Cpt(j, 1) - Wsize(1) ) || ( row >= Cpt(j, 1) + Wsize(1) ) || ...
```

```
            (col <= Cpt(j,2)- Wsize(2)) || (col >= Cpt(j,2) + Wsize(2) ) ;
```

```
        if ~valid
```

```
            break;
```

```
        end
```

```
    end
```

```
    if valid
```

```
        Cpt = [Cpt; [row, col]];      Fpt = Fpt + 1
```

```
    end
```

```
    i = i + 1;
```

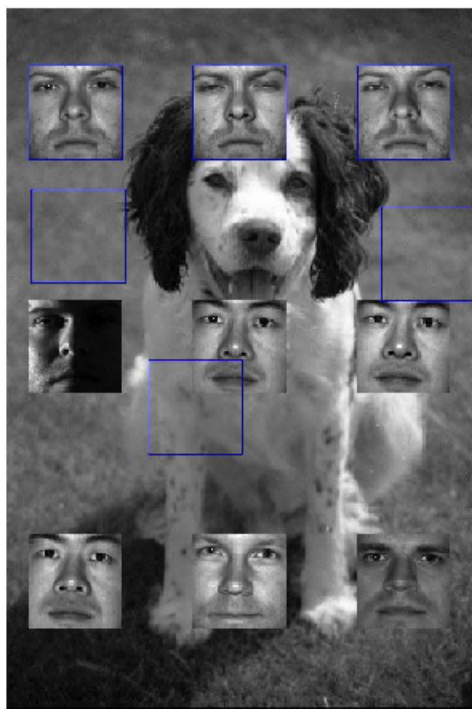
```
end
```

```
imshow(imread('Image.tiff')) ; hold on;
```

```
for i = 1 : Fpt
```

```
    rectangle('Position', [Cpt(i,2), Cpt(i,1), Wsize(2), Wsize(1)], 'EdgeColor', 'blue');
```

```
end
```



This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.