Advanced Correlation Filters and MACH

Haroon Arshad

Nazeer Ahmed Khan

Advanced Correlation Filters

- Advanced correlation filters (CFs) were introduced over three decades ago to offer distortion-tolerant object recognition
- Mostly used for template matching.
- Correlation filters (CFs) have been successfully applied to a variety of pattern recognition applications

Advanced Correlation Filters

- CFs are used in applications such as
  - automatic target recognition (ATR)
  - autonomous surveillance
  - security applications
  - biometric recognition
    - face localization, face tracking, biometric encoding, fingerprint recognition, iris recognition , pedestrian action recognition

Correlation Filter Design

- CFs can be designed to yield correlation peaks for each target of interest in the scene
- Most CFs are templates carefully designed using a set of training images that captures the expected distortions (in testing), while exhibiting low response to clutter and background.
- Attractive properties such as shift-invariance, noise robustness and efficient implementation make CFs well-suited for ATR and biometric applications
- **Not Scale Invariant**

Correlation Filter Design

- A filter is designed using ocular images of the same subject
- A test image is correlated with the filter. If the test image (e.g., a face) is from the same subject, the correlation plane should exhibit a sharp peak at the ocular region, otherwise the correlation plane should not have any significant peak.
- This peak or some other relevant metric such as the **peak-to-correlation energy (PCE) or peak-to-sidelobe ratio (PSR)** is used to determine whether the query image is from the authentic class or not.
- The location of the peak indicates the location of the target in the image.

Correlation Filter Implementation

- In the CF approach, a carefully designed template (loosely called a **filter** and in some literature a **sliding window template**) h(m,n) is cross-correlated with a query image x(m,n) to produce the output g(m,n).
- This operation is efficiently carried out in the frequency domain,

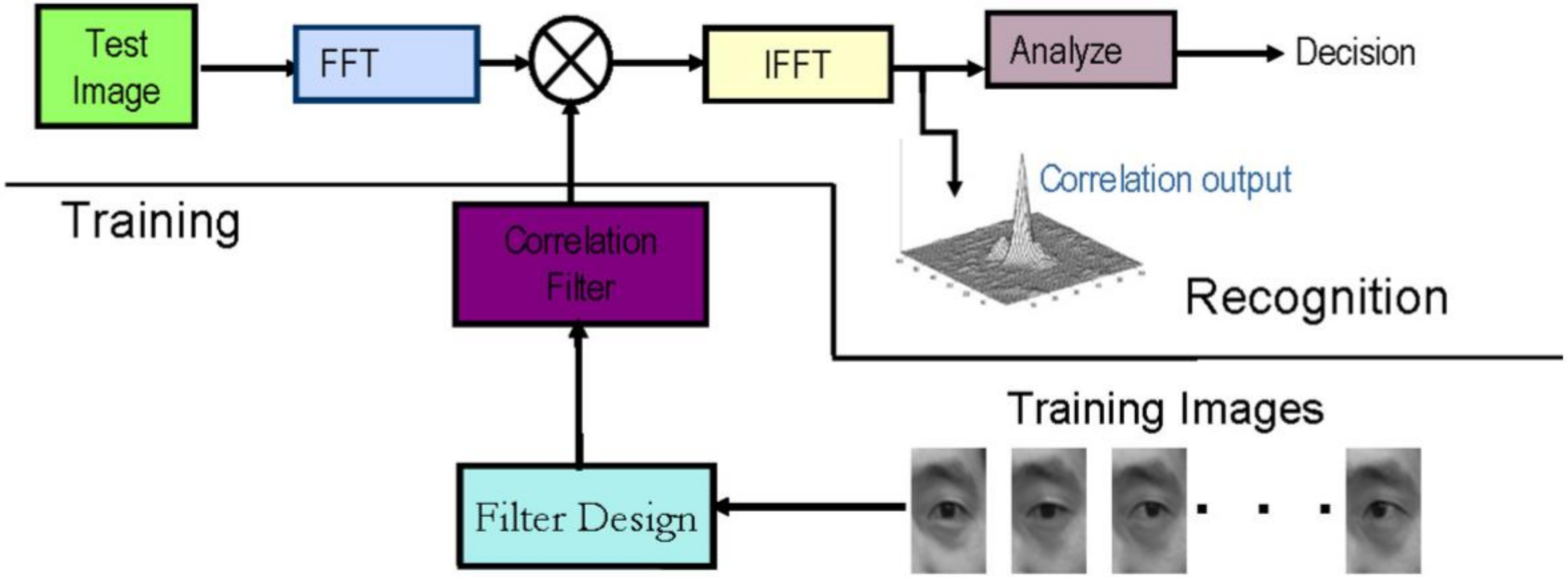
$$G(u,v) = X(u,v),H^*(u,v)$$

- G(u,v), X(u,v) and H(u,v) are the 2-D discrete Fourier transforms (DFTs) of the correlation output, the query image, and the template, respectively.

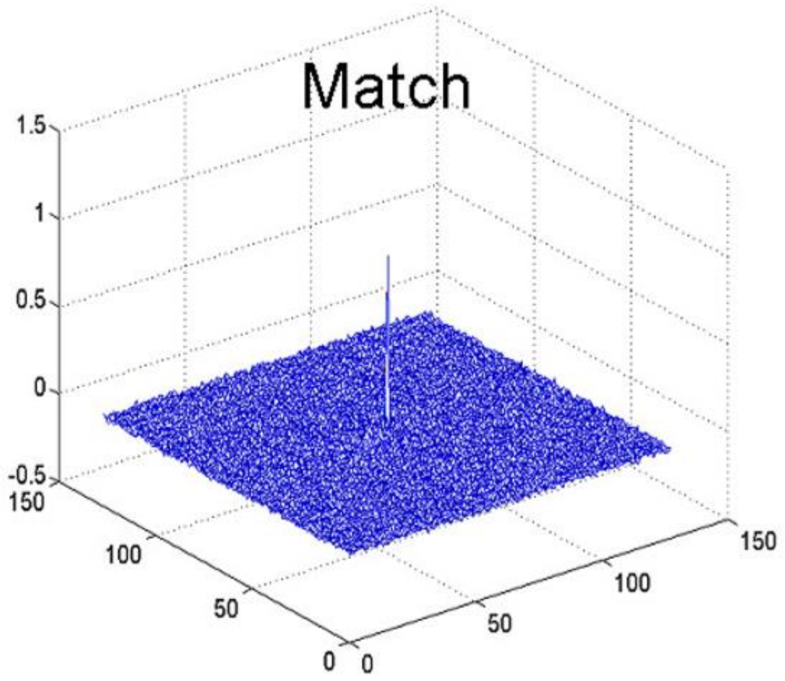
Correlation Filter Implementation

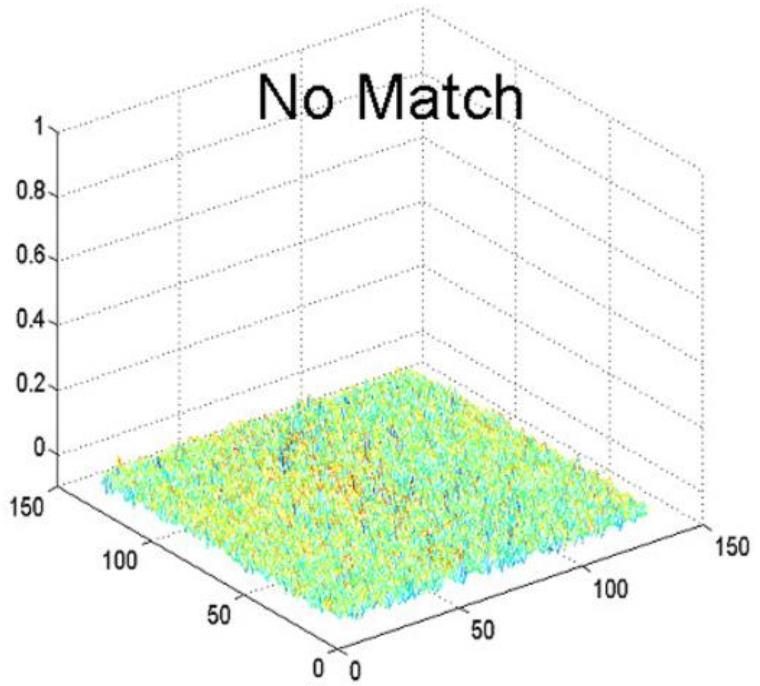
- When the query image is an authentic match (also called true-class or Class-1), g(m,n) should exhibit a sharp peak at the center of the biometric signature's location
- when the query image is an impostor (also called false-class or Class-2) g(m,n) should not have any significant peak
- The higher the correlation peak, the higher the probability that the image is authentic
- CFs offer the ability to simultaneously localize and classify an object of interest

Correlation Filter Implementation



Correlation Filter Design Result





Advanced Correlations Filters

- Some of the advances in CF design include
  - minimum average correlation energy (MACE)
  - optimal tradeoff synthetic discriminant function (OTSDF) filters
  - maximum average correlation height (MACH) filter
  - quadratic correlation filters (QCFs)
  - maximum margin correlation filters (MMCFs)
  - zero-aliasing correlation filters (ZACFs)

Minimum average correlation energy (MACE)

- MACE filter is designed to reduce the energy of the correlation output resulting in a sharp peak at the location of the target facilitating target recognition
- The MACE filter facilitates recognition by producing very sharp delta-function-like peaks with minimum sidelobes for authentic class training images and no such sharp peaks for imposter training images

Minimum average correlation energy (MACE)

$$h = D-1X(X^*D-1X)-1u$$

where diagonal matrix $D_i = X_i . X^*_i$ contains the power spectrum of X_i along its diagonal, and diagonal matrix D contains the average power spectral density of the training images along its diagonal

Maximum Average Correlation Height

- The first unconstrained CFs introduced in 1994 were the maximum average correlation height (MACH) filter and the unconstrained MACE (UMACE) filter.
- Previous CFs were constrained to produce an inner product of h*x = u for the training images, but such hard constraints are not necessarily satisfied by the non-training images.
- Removing these constraints increases the solution space and may improve the chances of finding a filter with better recognition performance.

Maximum Average Correlation Height

- The MACH filter is designed to minimize the average (dis-)similarity measure (ASM), i.e., the scatter of the correlation planes, and simultaneously minimize the ACE and maximize the average correlation peak intensity .
- The MACE is a variations of the MACH filter that ignores ASM.
- Filter Size will be equal to Training Images size

$$H = \frac{\text{Training Images Mean}}{\alpha.C + \beta.D + \gamma.Sx}$$

Maximum Average Correlation Height

$$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{X}_i - \text{Image Mean})(\mathbf{X}_i - \text{Image Mean})^*$$

$$D = |hx|^2 = \frac{1}{N}\sum_{i=1}^{N} |\text{Image}_i \,.\, \text{Image}^*_i|^2$$

C = Max (Diagonal D)

C = average correlation peak intensity

Maximum Average Correlation Height (Training Parameters/ Controlling Parameter)

$\alpha =$ experimentally that parameter is found to control the discriminability of the filter;the smaller the value, the less ability of discriminating

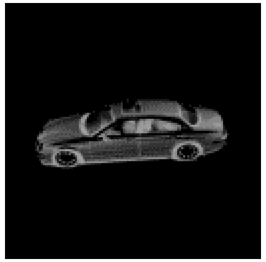
$\beta =$ experimentally that parameter is found to control the sidelobes of the output correlation plane

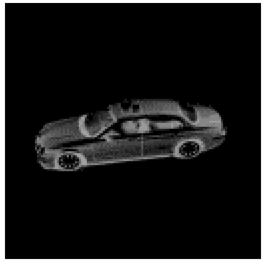
with other words it controls the distortion tolerance! The smaller the value, the sharper the peak gets

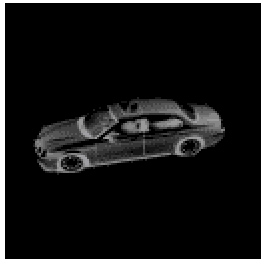
$\gamma =$ 0.01 Fix

Example – To find a car in a Scene

Training Images







```
refImages{1} = rgb2gray(imread('training (1).jpg'));
refImages{2} = rgb2gray(imread('training (2).jpg'));
refImages{3} =  rgb2gray(imread('training (3).jpg'));

[denom img_mean] =
MachDenominator(refImages,a, b, y);

H = conj(img_mean) / denom;
```

Example – To find a car in a Scene

- **Calculate training Images Mean**
  - for i = 1:N
  - fft_Train_img = sfft(refImages{i});
  - sum_Train_img = double(sum_Train_img + fft_Train_img);
  - end
  - img_mean = double(sum_Train_img./N);

Calculate ASM (Average Similarity Measure)

- fft_Train_img = sfft(refImages{i});
- Calculate the Similarity matrix of the training images
  - SS = double(conj(fft_Train_img - img_mean).*(fft_Train_img - img_mean));
  - Formula_Sim_Sum = double(Formula_Sim_Sum + SS);

Calculate Average Correlation Energy (ACE)

- DD = double((conj(fft_Train_img)).*(fft_Train_img));
-     Formula_Diag_Sum = double(Formula_Diag_Sum + DD);

- C3 = max(max(D));

Final MACH Filter Equation

- `H = conj(img_mean) ./ denom;`
  - **denom = double((b.*D)+(a.*C3)+(y.*S));**

Now apply MACH Filter on testing Image (Scene Image)

```
for i = 1 : N
    CmplxMult = double(Refconj .* Input_fft);
end
Surf(CmplxMult );
```

Result

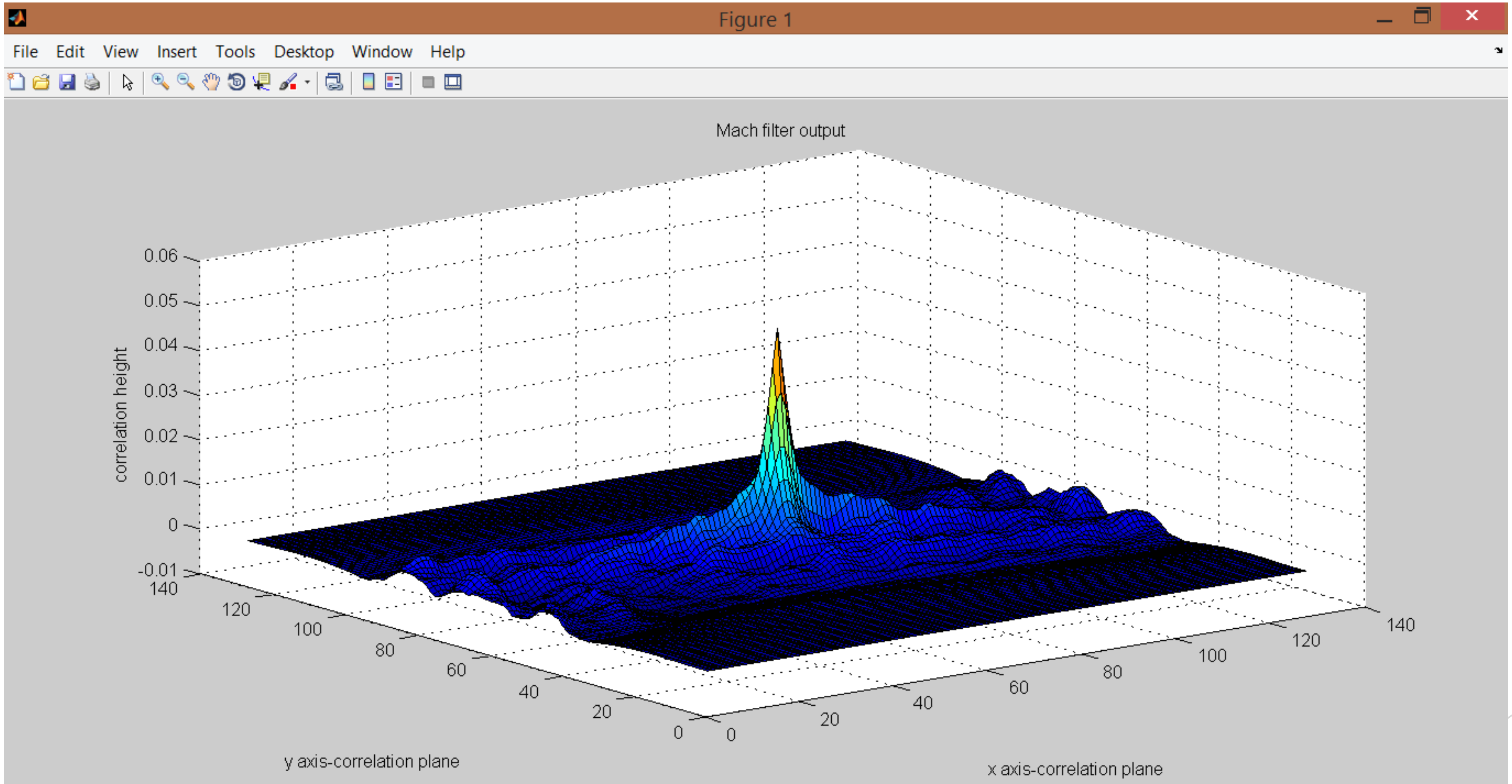
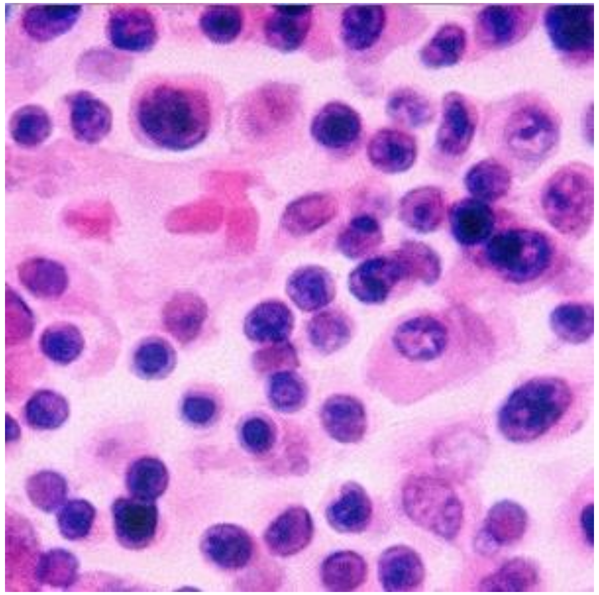


Figure 1
File Edit View Insert Tools Desktop Window Help
Mach filter output
correlation height
0.06
0.05
0.04
0.03
0.02
0.01
0
-0.01
140
120
100
80
60
40
20
0
y axis-correlation plane
0
20
40
60
80
100
120
140
x axis-correlation plane

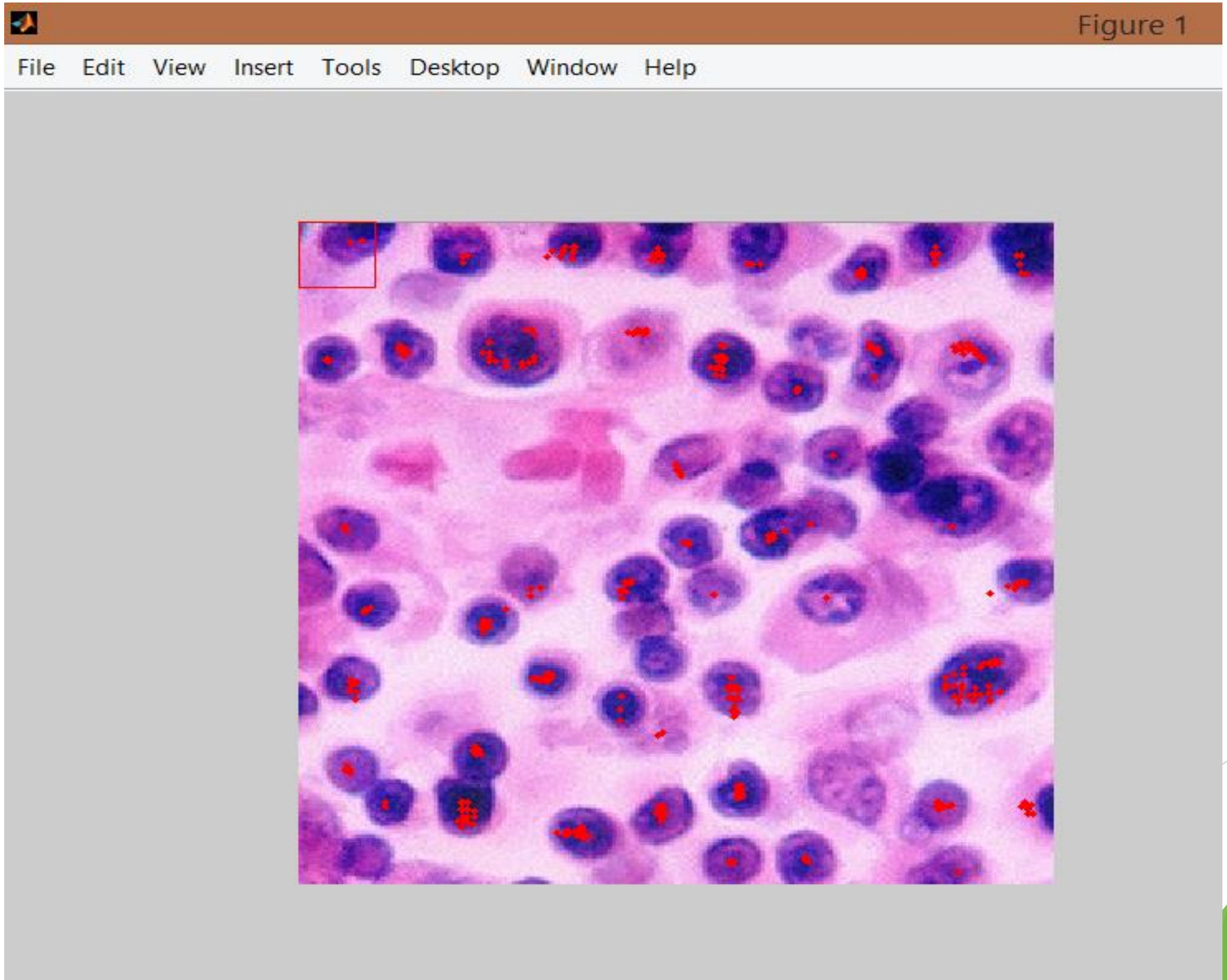
Example 2 (To detect and count Mirco Cells

Training Images

Test Image



Example 2 (To detect and count Mirco Cells



Thanks