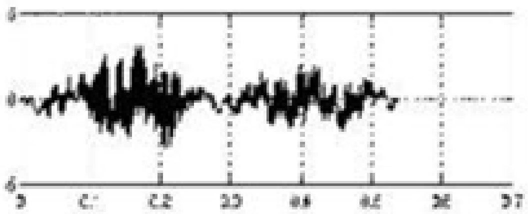
PATTERN RECOGNITION

Fitri Utaminingrum

WHAT IS A PATTERN?

- A pattern is an abstract object, or a set of measurements describing a physical object.
- For example, a pattern could be
  - A fingerprint images
  - A handwritten cursive word
  - A human face
  - A speech signal



WHAT IS PATTERN RECOGNITION?

- The study of how machines can observe the environment,
  - learn to distinguish *patterns* of interest from their background.

EXAMPLES OF APPLICATIONS

Optical Character Recognition (OCR)

- **Handwritten**:
- **Printed texts**: reading machines for blind people

Biometrics

- **Face recognition, verification, retrieval.**
- **Finger prints recognition.**
- **Speech recognition.**

Diagnostic systems

- **Medical diagnosis**: X-Ray, EKG (ElectroCardioGraph) analysis.

Military applications

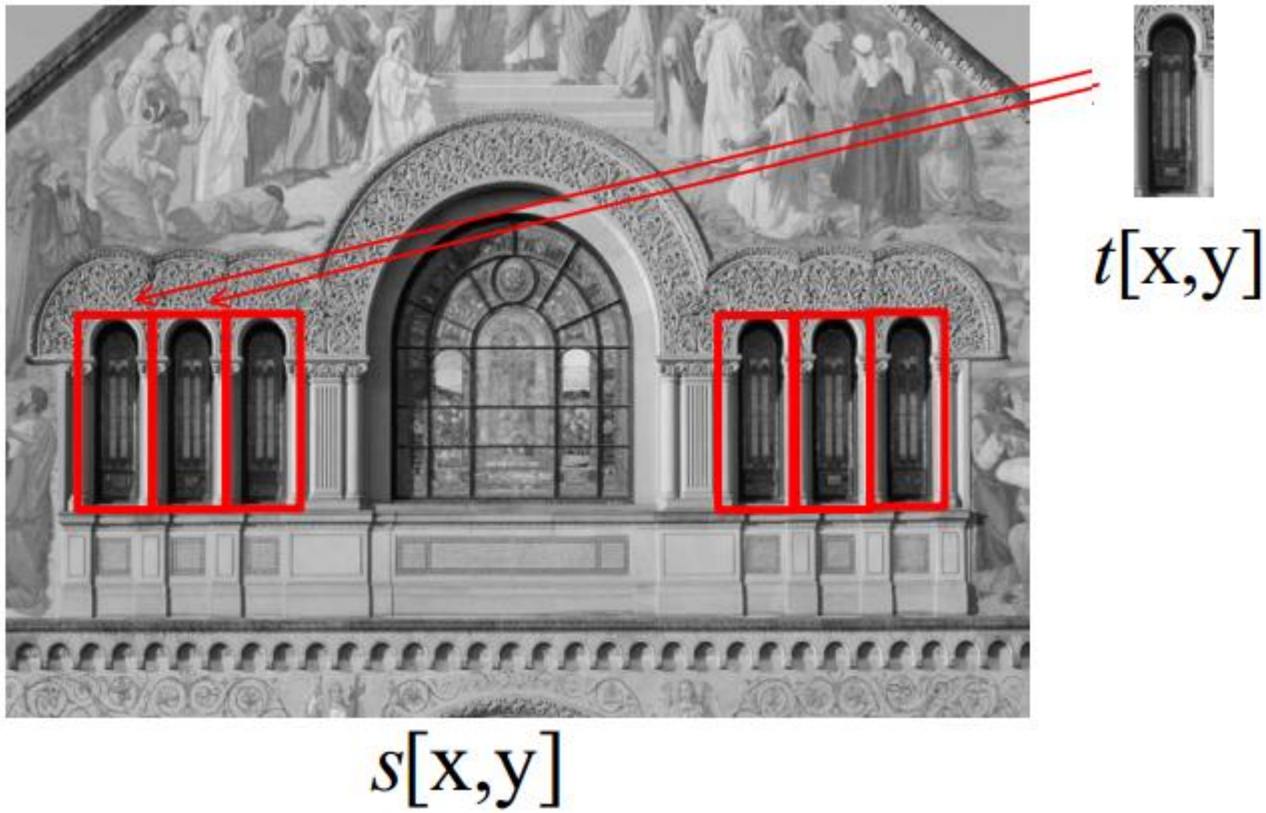
- **Automated Target Recognition** (ATR).
- **Image segmentation and analysis** (recognition from aerial or satelite photographs).

PATTERN RECOGNITION APPROACH

1. Template Matching
2. Statistical
3. Syntactic or Structural
4. Neural Network

TEMPLATE MATCHING

- Locate an object, described by a template t[x,y], in the image s[x,y]
- Example:



TEMPLATE MATCHING

- Sensitivity to changes in size and rotation
  - Handling variations in size
    1. Normalisation: Transform image to standard size

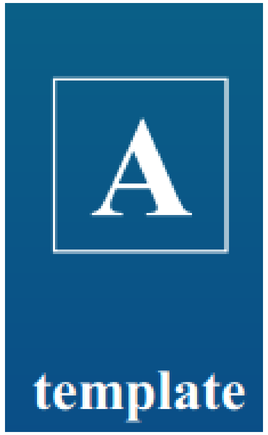
       Works only if there is no size variation within the image

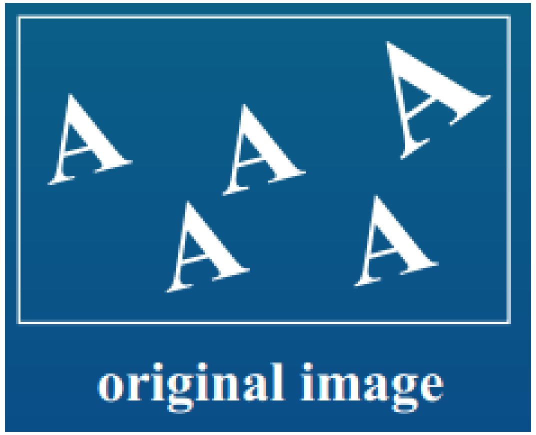
    2. Adaptivity: Spatially scale and rotate the template in each position, select the best matching scale and rotation

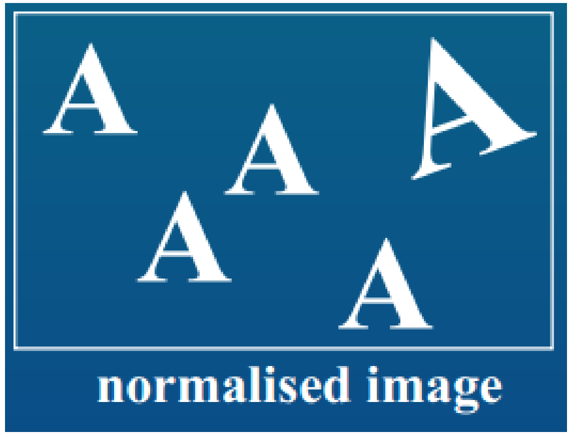
       Very slow if number of scales and rotations is large

       Used only for small number of scales and rotations

Normalising an image for size and orientation







- The letter A in the top right corner differs in size and orientation.
    - This letter will not match.
- The other four letters will match.

Template Matching

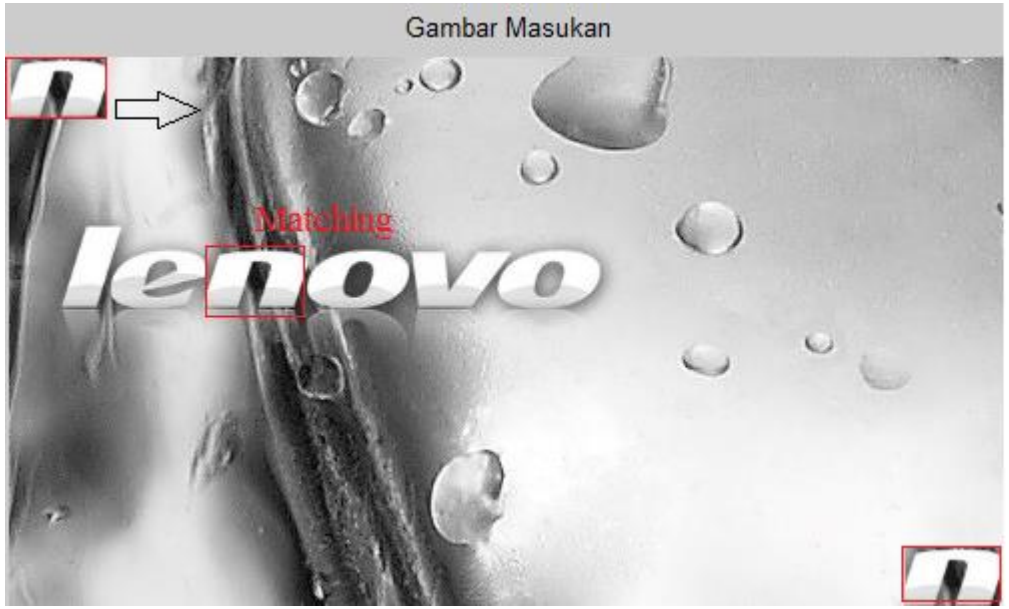
- Search for the best match by mean-squared error.

$$\mathrm{MSE} = \frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}\left(f(x,y) - f'(x,y)\right)^2$$

  - f(x,y) is region interest
  - f’(x,y) is template

Template matching method by MSE approach

- Based on the sliding template
- Calculated MSE in each position of template
- Determine the minimum MSE value tolerance for matching



Code Matlab

```
function [fo hit] =
TempMatchingMSE(fi,ft)

fi=double(fi)./255;

ft1=double(ft)./255;

[a1 b1] = size(fi(:,:,1));

[a2 b2] = size(ft1(:,:,1));

fo=fi;

k=1;
```

```
for x=1:a1
    for y=1:b1
     if (x)<=(a1-a2) && (y)<= (b1-b2)
        fa= fi(x:(x+a2-1),y:(y+b2-1),1:3);
        NilaiMSE = MSEku(fa(:,:,1),ft1(:,:,1));
        if NilaiMSE <= 0.045
           fo(x:x+a2-1,y,1)=1;
           fo(x:x+a2-1,y,2)=0;
           fo(x:x+a2-1,y,3)=0;
             fo(x:x+a2-1,y+b2,1)=1;
             fo(x:x+a2-1,y+b2,2)=0;
             fo(x:x+a2-1,y+b2,3)=0;
           fo(x,y:y+b2,1)=1;
           fo(x,y:y+b2,2)=0;
           fo(x,y:y+b2,3)=0;
             fo(x+a2,y:y+b2,1)=1;
             fo(x+a2,y:y+b2,2)=0;
             fo(x+a2,y:y+b2,3)=0;
        end
    end
        hit(k)=NilaiMSE;
        k=k+1;
    end
end
return
```

Template Matching

- Search for the best match by Correlation

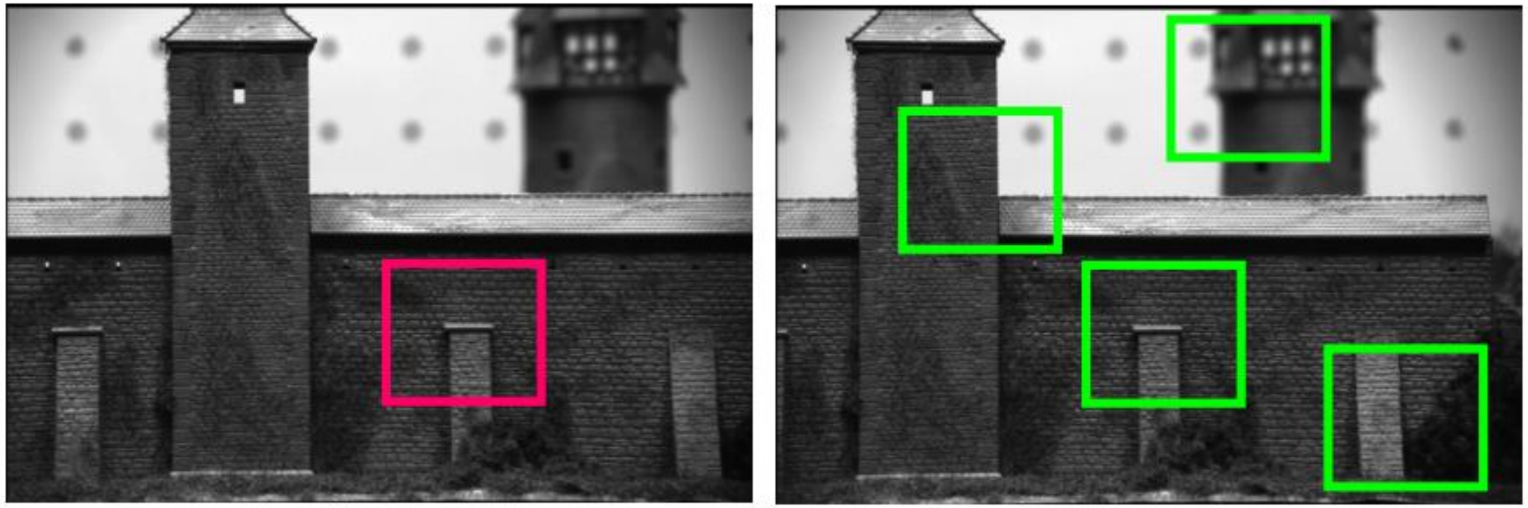
$$\mathrm{r} = \frac{\sum_{y=1}^{M}\sum_{x=1}^{N}(f(x,y)-\bar{f})(f_T(x,y)-\bar{f_T}}{\sqrt{AB}}$$

$$\mathrm{A} = \sum_{y=1}^{M}\sum_{x=1}^{M}(f(x,y)-\bar{f})$$

$$\mathrm{B} = \sum_{y=1}^{M}\sum_{x=1}^{N}(f_T(x,y)-\bar{f_T})$$

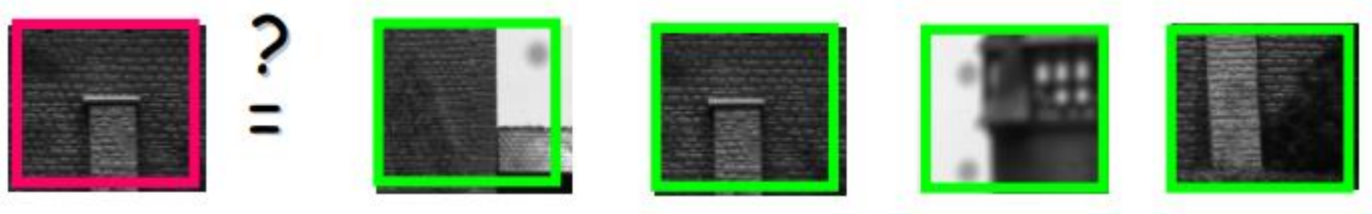
Template matching method by Correlation approach

- cut little pictures out from an image, then tried convolve them with the same or other images
- Elements to be matched are image patches of fixed size



Template matching method by Correlation approach

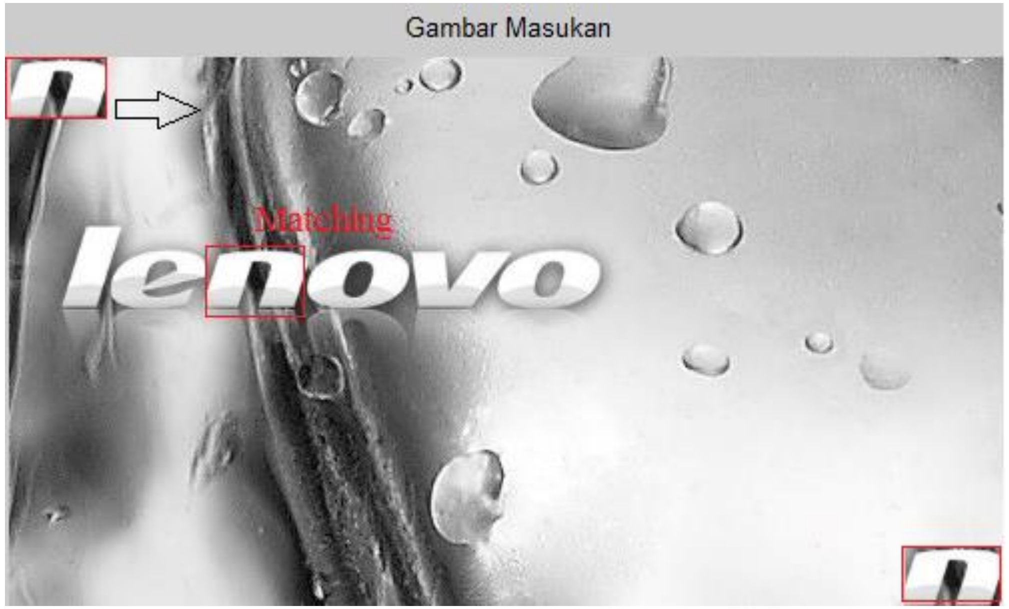
Task: what is the corresponding patch in a second image?



- Need an appearance similarity function.
- Need a search strategy to find location with highest similarity. Simplest (but least efficient) approach is exhaustive search

Template matching method by Correlation approach

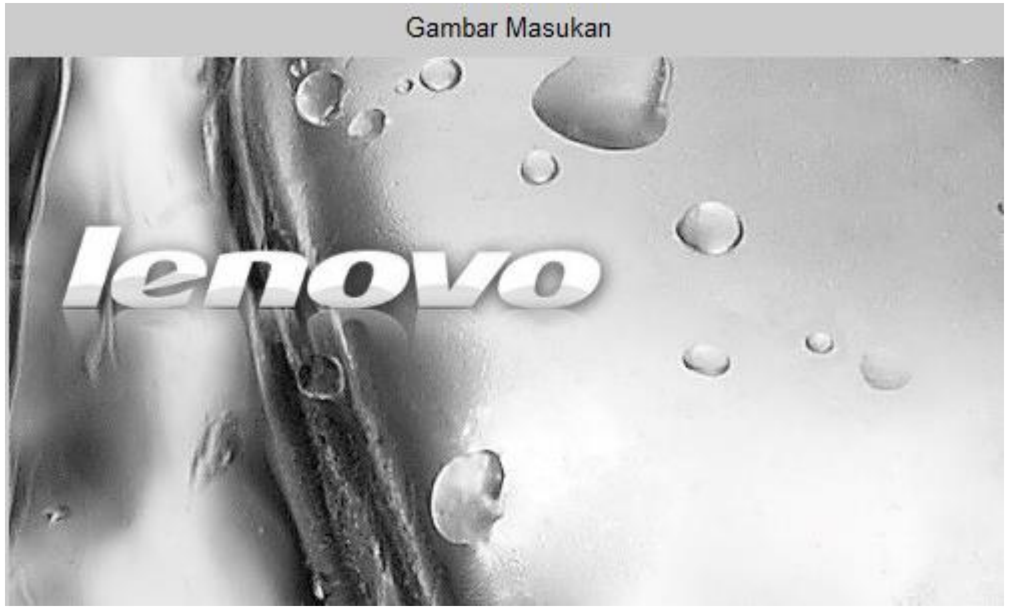
- Based on the sliding template
- Calculated correlation in each position of template
- Determine the correlation value tolerance for matching

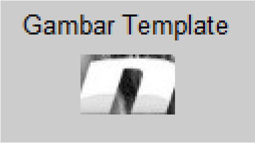


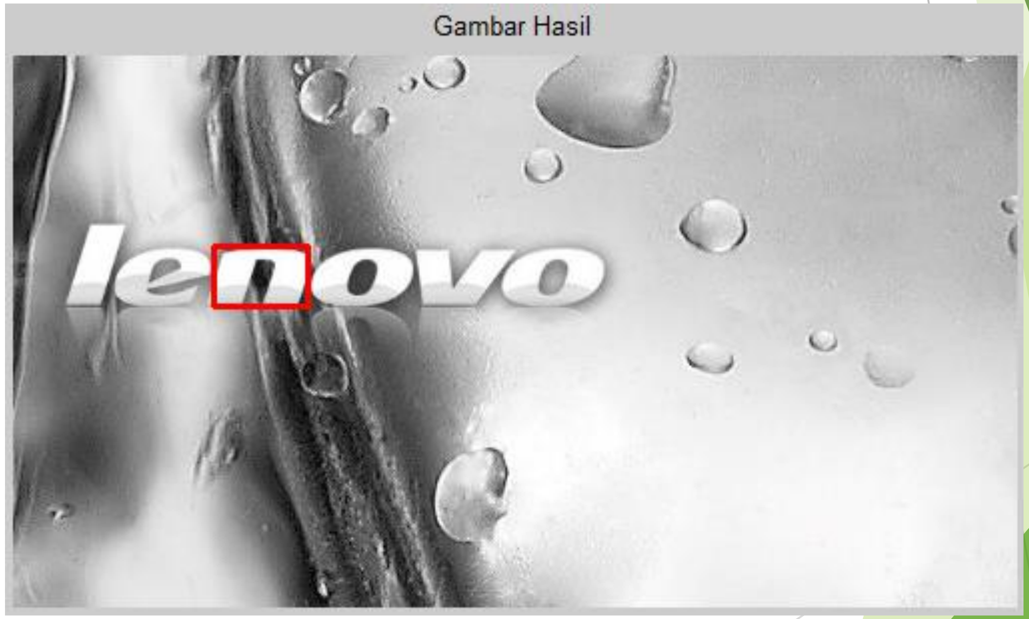
Example

```
function [fo hit] = TempMatchingKu(fi,ft)
fi=double(fi)./255;
ft=double(ft)./255;
[a1 b1] = size(fi(:,:,1));
[a2 b2] = size(ft(:,:,1));
fo=fi;
k=1;
for x=1:a1
   for y=1:b1
    if (x+a2-1)<=(a1-a2) && (y+b2-1)<= (b1-b2)
      fa= fi(x:(x+a2-1),y:(y+b2-1),1:3);
      NilaiCor = corr2(fa(:,:,1),ft(:,:,1));
      if NilaiCor >= 0.89
        fo(x:x+a2-1,y,1)=1;   fo(x:x+a2-1,y,2)=0;   fo(x:x+a2-1,y,3)=0;
        fo(x:x+a2-1,y+b2,1)=1; fo(x:x+a2-1,y+b2,2)=0;  fo(x:x+a2-1,y+b2,3)=0;
        fo(x,y:y+b2,1)=1;  fo(x,y:y+b2,2)=0;   fo(x,y:y+b2,3)=0;
        fo(x+a2,y:y+b2,1)=1;  fo(x+a2,y:y+b2,2)=0;  fo(x+a2,y:y+b2,3)=0;
      end
   end
      hit(k)=NilaiCor;
      k=k+1;
   end
end
return
```

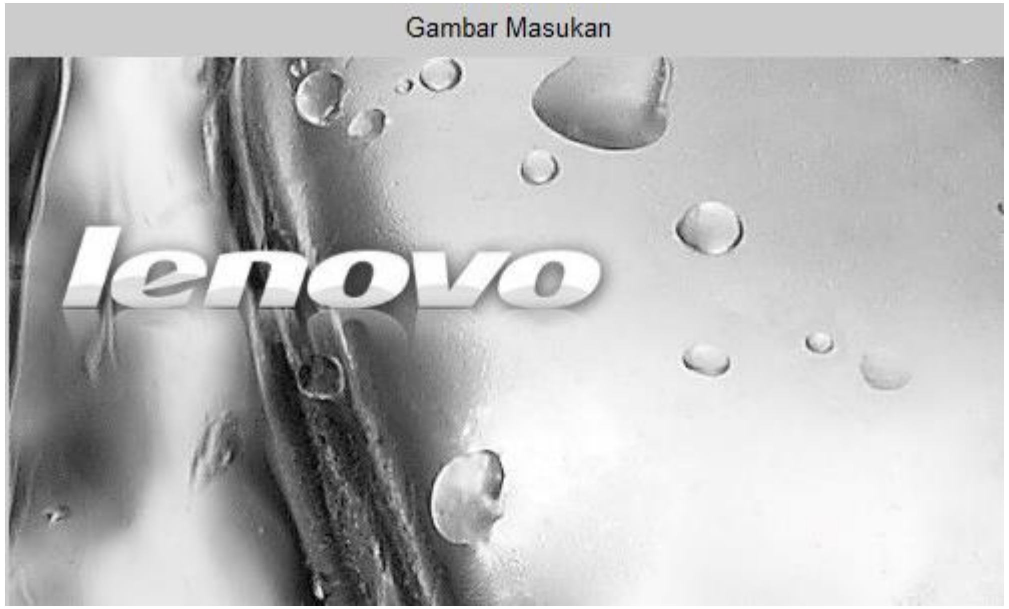
Hasil running dengan template n

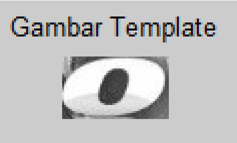


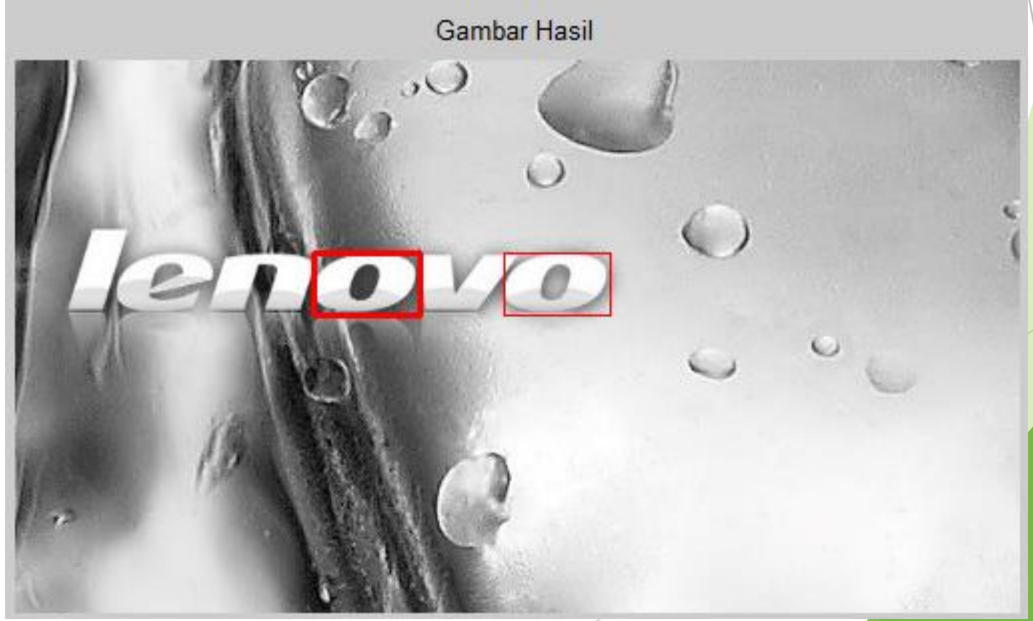




Hasil running dengan template o

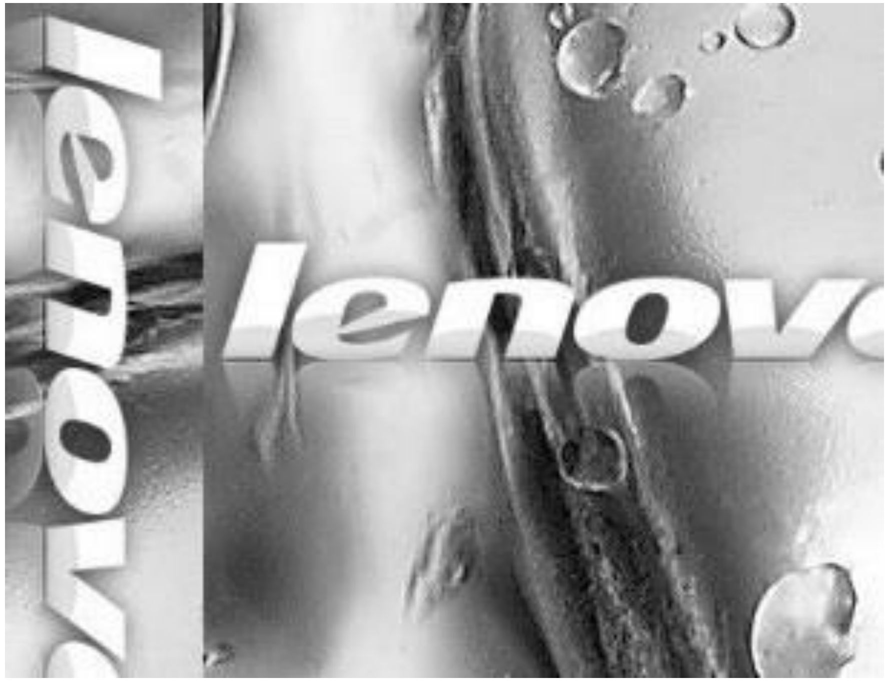






Case study

- Improve code Matlab for the problem of image below !



Problem with Correlation of Raw Image Templates

- Consider correlation of template with an image of constant grey value:

$$\begin{array}{|c|c|c|}\hline a & b & c \\ \hline d & e & f \\ \hline g & h & i \\ \hline\end{array} \otimes \begin{array}{|c|c|c|}\hline v & v & v \\ \hline v & v & v \\ \hline v & v & v \\ \hline\end{array}$$

- Result: v X (a+b+c+d+e+f+g+h+i)

Problem with Correlation of Raw Image Templates

- Now consider correlation with a constant image that is twice as bright.

$$\begin{array}{|c|c|c|}\hline a & b & c \\ \hline d & e & f \\ \hline g & h & i \\ \hline \end{array} \otimes \begin{array}{|c|c|c|}\hline 2v & 2v & 2v \\ \hline 2v & 2v & 2v \\ \hline 2v & 2v & 2v \\ \hline \end{array}$$

- Result: 2 X v X (a+b+c+d+e+f+g+h+i)

  > vX(a+b+c+d+e+f+g+h+i)

- Larger score, regardless of what the template is!

Solution

- Subtract off the mean value of the template.
- In this way, the correlation score is higher only when darker parts of the template overlap darker parts of the image, and brighter parts of the template overlap brighter parts of the image.

“SSD ” or “block matching ” (Sum of Squared Differences)

$$\sum_{[i,j]\in R} (f(i,j) - g(i,j))^2$$

- The most popular matching score.
- We used it when deriving Harris corners
- T&V claim it works better than cross-correlation