SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA
THANJAVUR | KUMBAKONAM | CHENNAI

ENHANCEMENT OF SECURITY IN VIDEO COMMUNICATION THROUGH VISUAL CRYPTOGRAPHY AND FIREFLY OPTIMIZATION

TEAM MEMBERS :
1.ADITI E 123003006, CSE
2.BHAVYA BHARATHI 123015020,IT
3.BYSANI VENKATA LAKSHMI SAI JYOTHIKA 123015125,IT

GUIDE :
BHASKARAN S
Asst. Professor- II

OUTLINE

- It is to ensure and improve secure communication of the video frames through hidden water-marking which is implemented using cryptography.
- The video frames are extracted using firefly algorithm based on certain threshold value after which watermark is processed using visual cryptography.
- The data is embedded into video frames using a reversible data hiding procedure.
- The watermark is extracted after performing various types of attacks.

INTRODUCTION

The high exchange of information in various forms has brought new challenges in protecting data.

Attackers are trying to get access to secret information and this is also part of a violation of privacy.

To protect confidential data, reversible data hiding techniques that are histogram-based are

mostly used.

The histogram bit shifting method is a technique that is used to achieve reversible data hiding in images.

A watermark is embedded in the image in a secured manner.

The work in this project is based on using this technique for video communication.

The quality of the frame is defined by some parameters and there is a quality threshold for the selection of frames.

This threshold value is measured using the firefly optimization algorithm.

WORKPLAN

AIM: The aim is to achieve high security and video quality compared to other reversible data hiding techniques.

OBJECTIVES: The video frame should be able to accommodate and embed high capacity data and should be able to recover them with minimum distortion

- To understand and implement firefly optimization.
- To achieve good quality in the processing of water-marking symbol.
- To understand and implement embedding procedure
- To analyze data extraction algorithm

DATASET

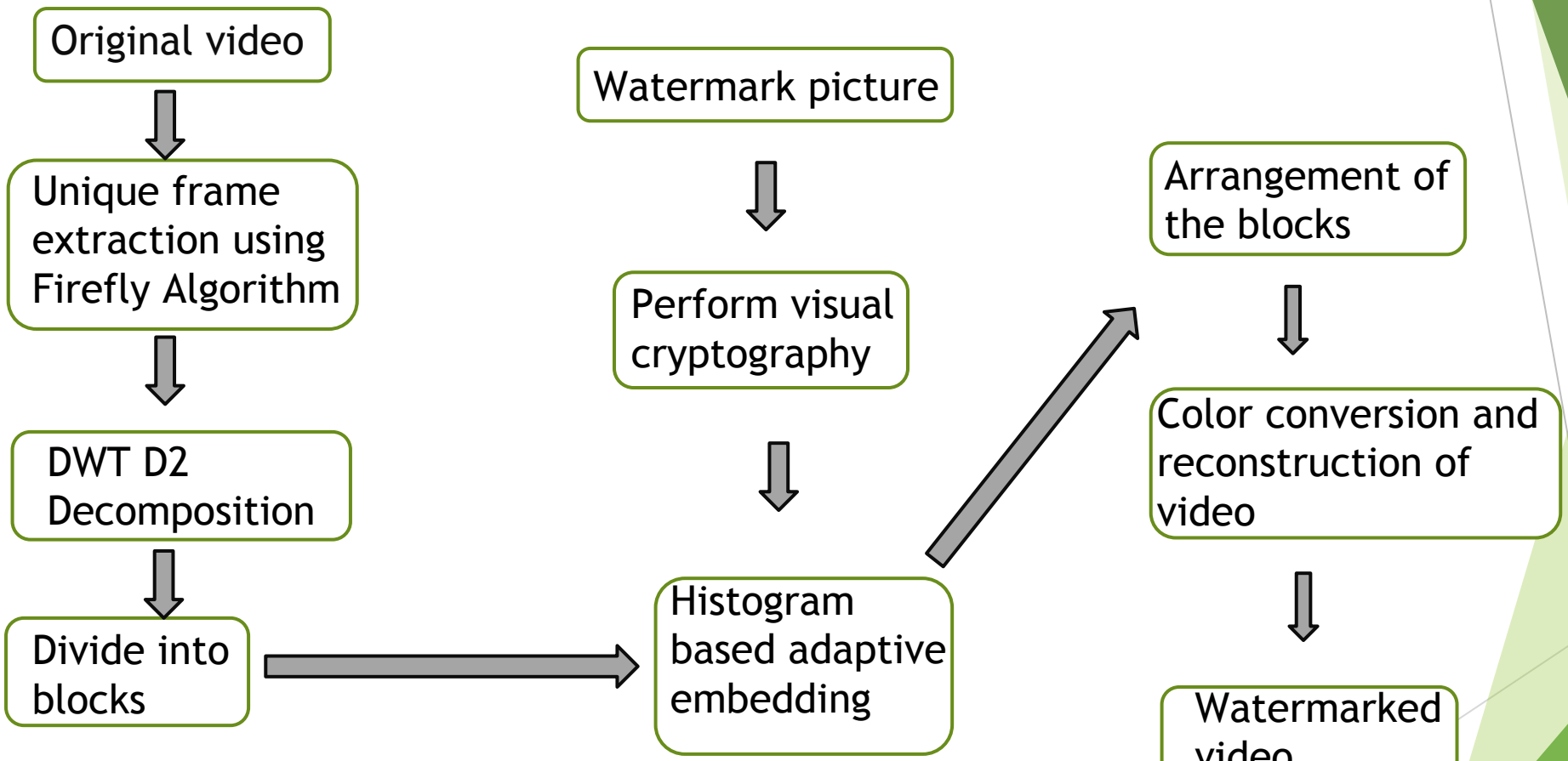
- LINK: https://github.com/wssmanojkumar/Watermaking-Sample-datset

DATASET DESCRIPTION

- This dataset is used to validate and ensure quality of the watermarked video and BER(Bit Error Rate) is used to test the robustness against various attacks.
- The quality and robustness of the approach is estimated using performance parameters such as peak signal-to-noise-ratio (PSNR) and bit error rate (BER).

Videos					
Images	 Case (1)	 Case (2)	Case (3)	 Case (4)	 Case (5)

WORKFLOW ARCHITECTURE



EXPECTED OUTCOME

- To achieve high security and video quality compared to other reversible data hiding techniques.
- The video frame should be able to accommodate and embed high capacity data and should be able to recover them with minimum distortion.

RELATED WORKS :

Agilandeswari and ganesan (2016) High robust watermarking scheme	Plane slicing based watermarking a embed coloured watermarking images on the colour video using Discrete Wavelet Transform(DWT), Singular value decomposition(SVD). High level of robustness watermark can survive even if the watermarked data go through distortions. High value of PSNR. Good fidelity- watermark is not visible to the user and it does not degrade the quality of the content.
Arab and karmakar (2016) Altering the frequency coefficients using DCT.	Discrete Cosine Transform(DCT) based rotation attack resistant video watermarking scheme. Algorithm also implemented in matlab and has been tested against three different standard videos. Resistant against any type of rotation attacks and video attacks.

Kulkarni and Kulkarni (2018) Cryptography-based greyscale image watermarking scheme.	This scheme was given to two shares of the images and finds out the results for three greyscale images. Satisfies - security, robustness and blindness.
Tang et al (2019) Reversible data hiding approach.	Huffman code- to reduce the size of embedding location maps. Not suitable for JPEG images but good for data hiding capacity and computational time.

FIREFLY OPTIMIZATION ALGORITH

1. Initialize parameters
2. Generate population of n fireflies
3. Calculate fitness value of each firefly
4. Check if(t:=1 to Maxt)
5. Update position and light intensity for each firefly
6. Report the best solution.

OBJECTIVE FUNCTION :

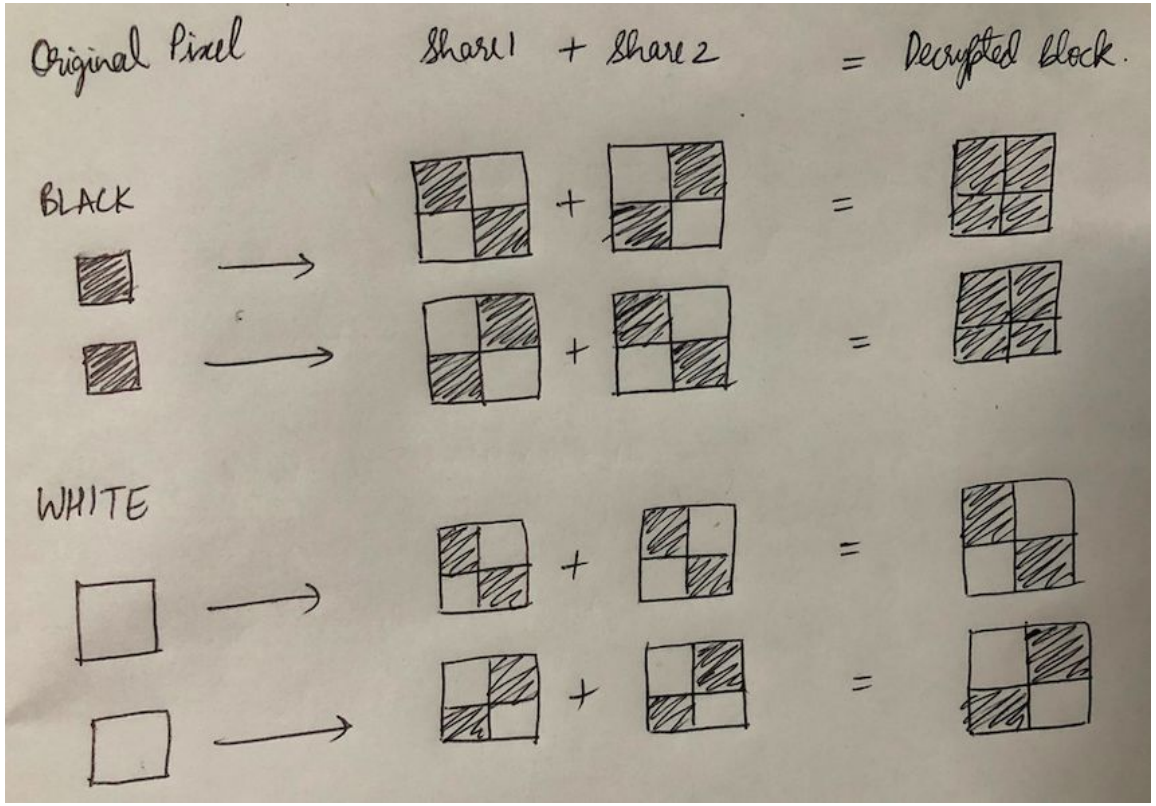
$$objective = PSNR + \emptyset * [BER(w, w') + \sum_{i=1}^{At} BER(w, w_i')]$$

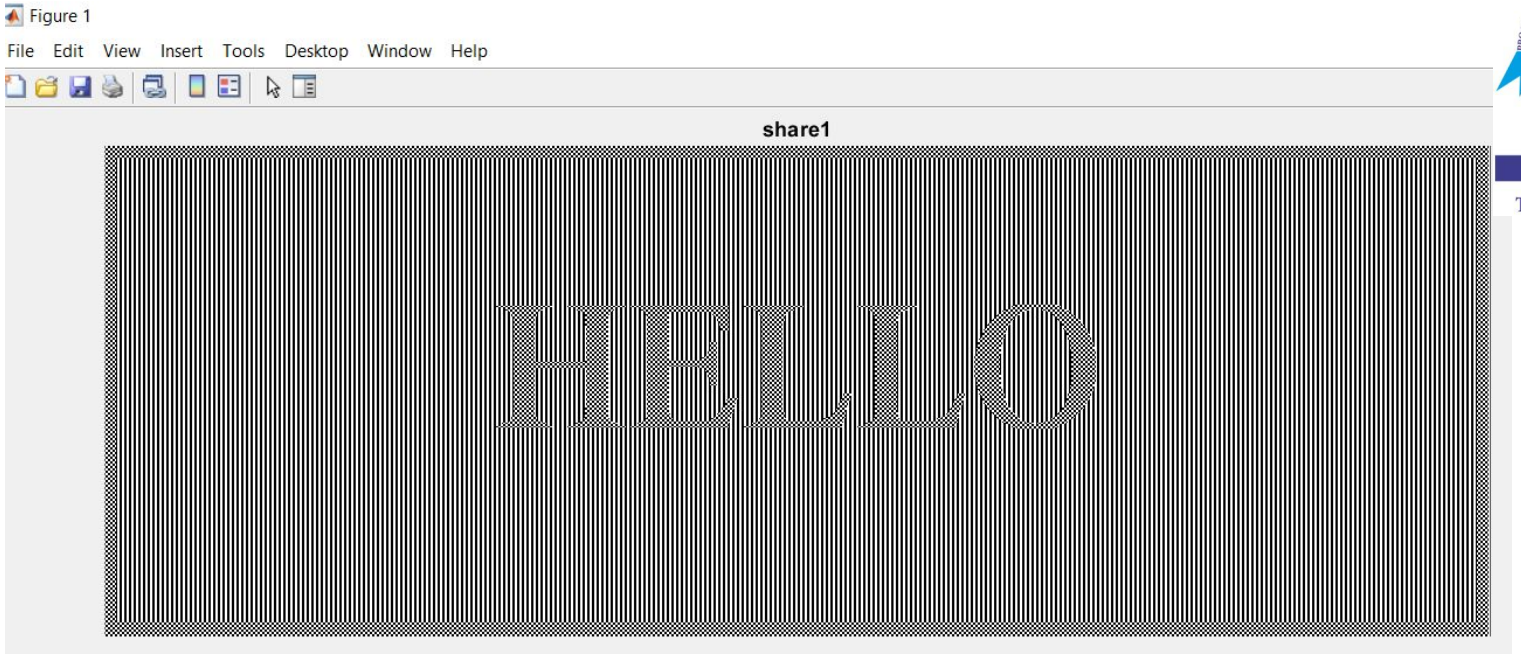
Based on the given objective function there are certain number of weights or parameters that are to be considered to calculate the threshold value based on which we select frames for embedding our watermark picture.

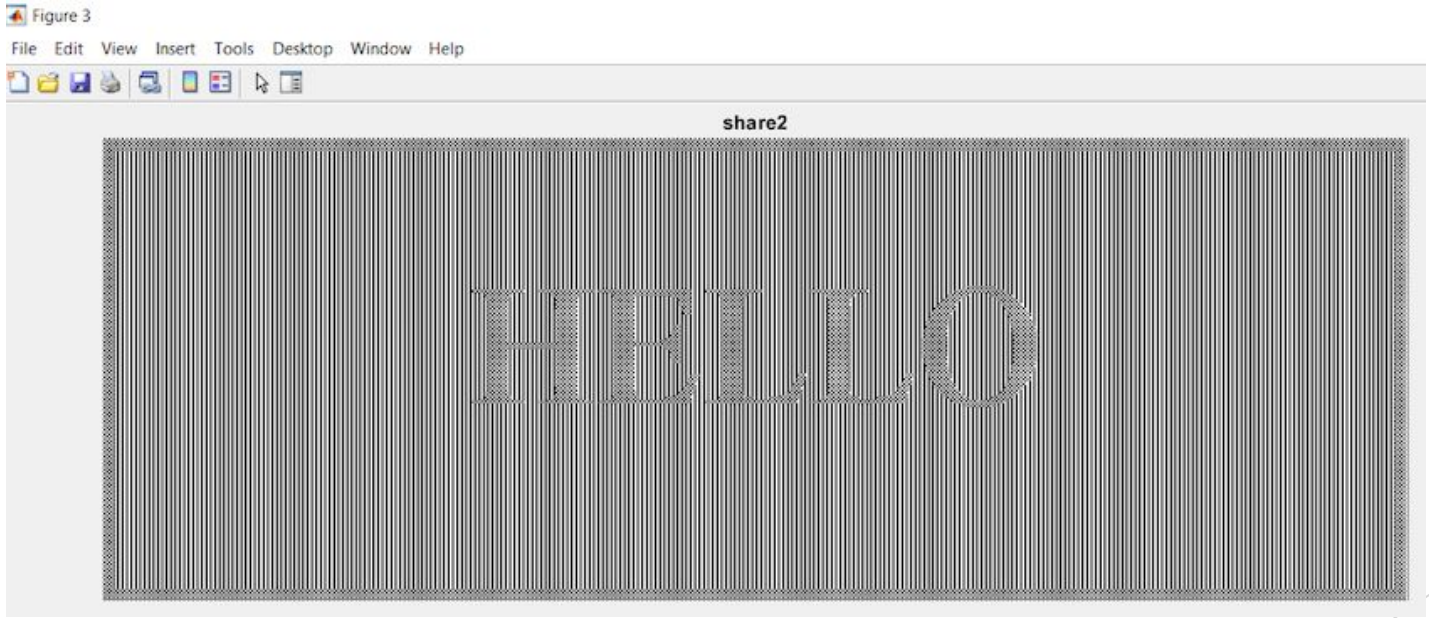
We are still exploring what those parameters are and how to optimize them into a single threshold or complexity value.

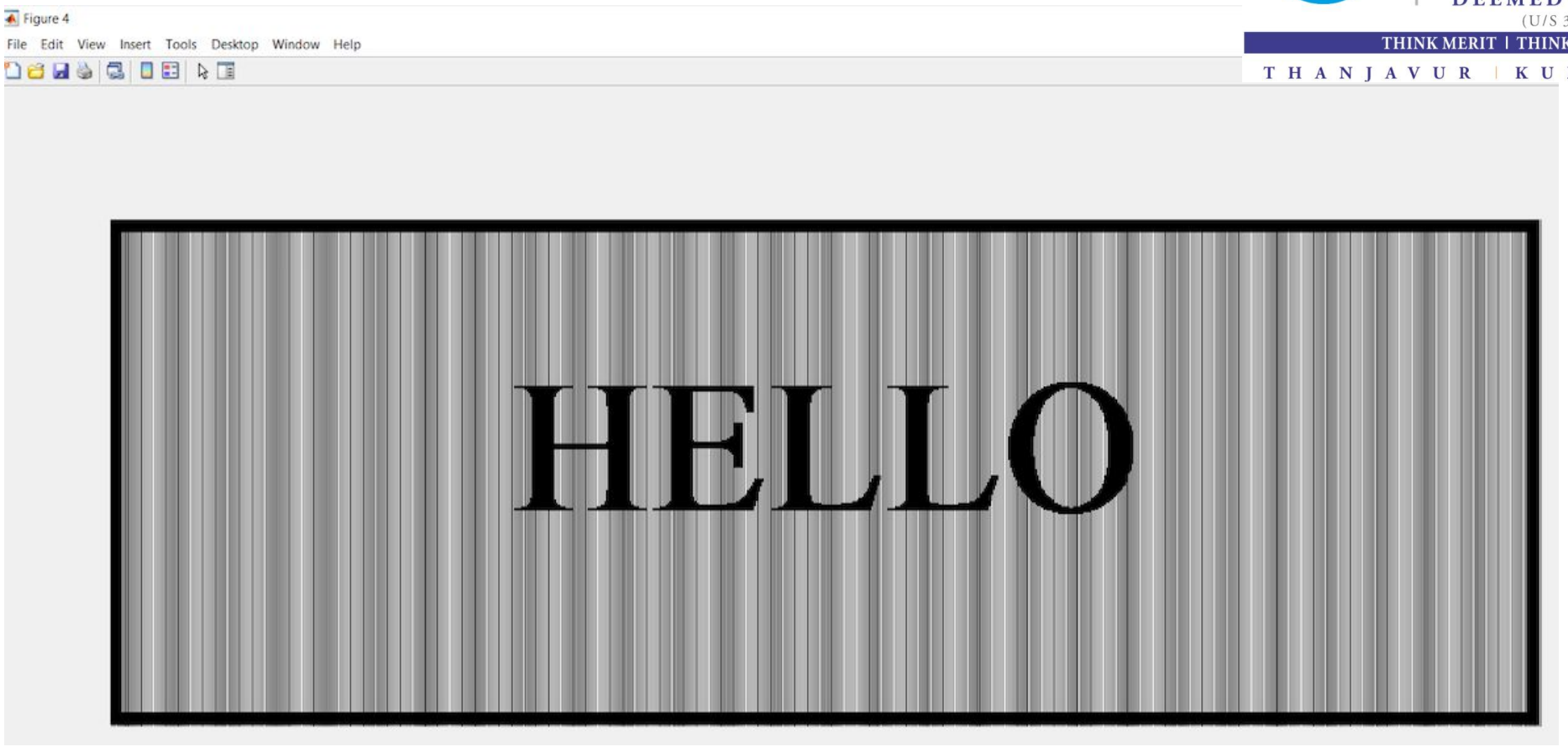
VISUAL CRYPTOGRAPHY

It’s an encryption technique on images or text in which decryption is done by human visual system.

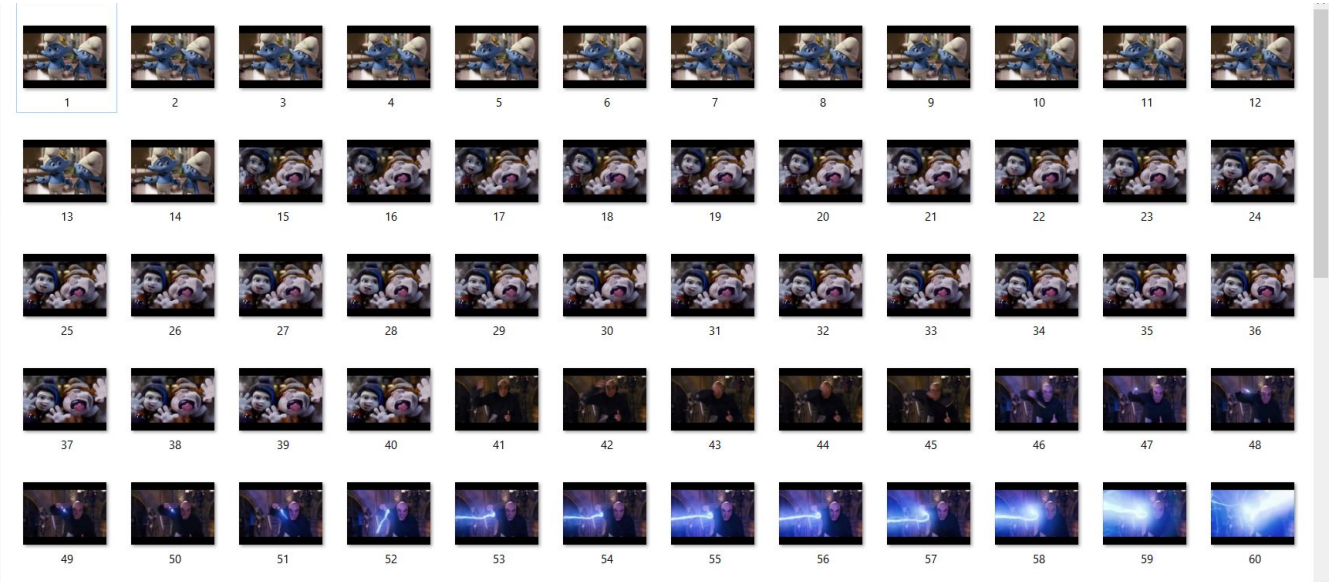


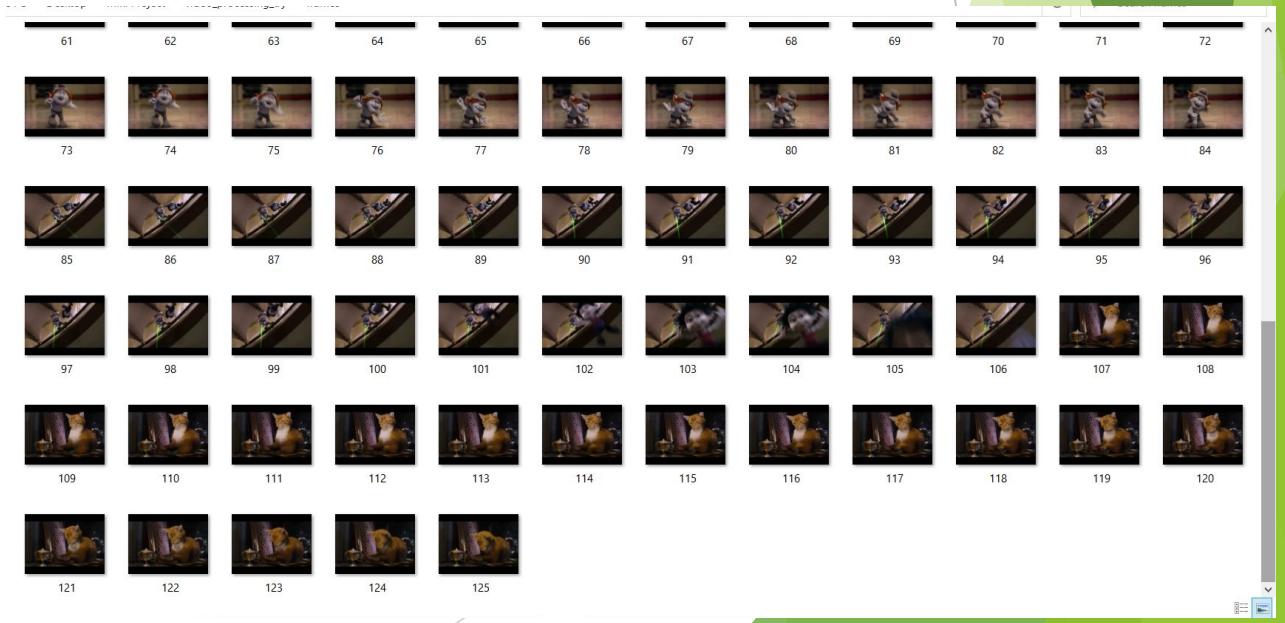






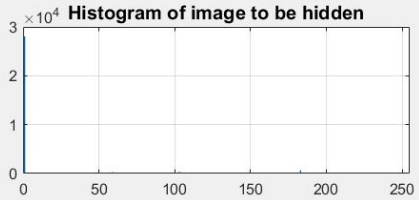
Processing a video and extraction of frames





Embedding and extracting watermark in one frame of the video

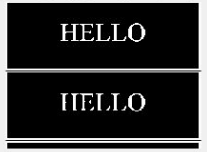
Image to be Hidden



Hidden Image Thresholded at 70

Original Grayscale Starting Image

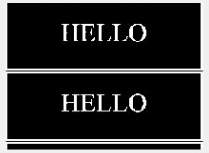
Hidden Image
to be Inserted into Bit Plane 3



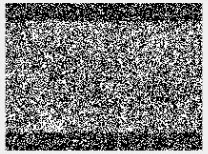
Final Watermarked Image
without added Noise

Watermarked Image
with added Noise

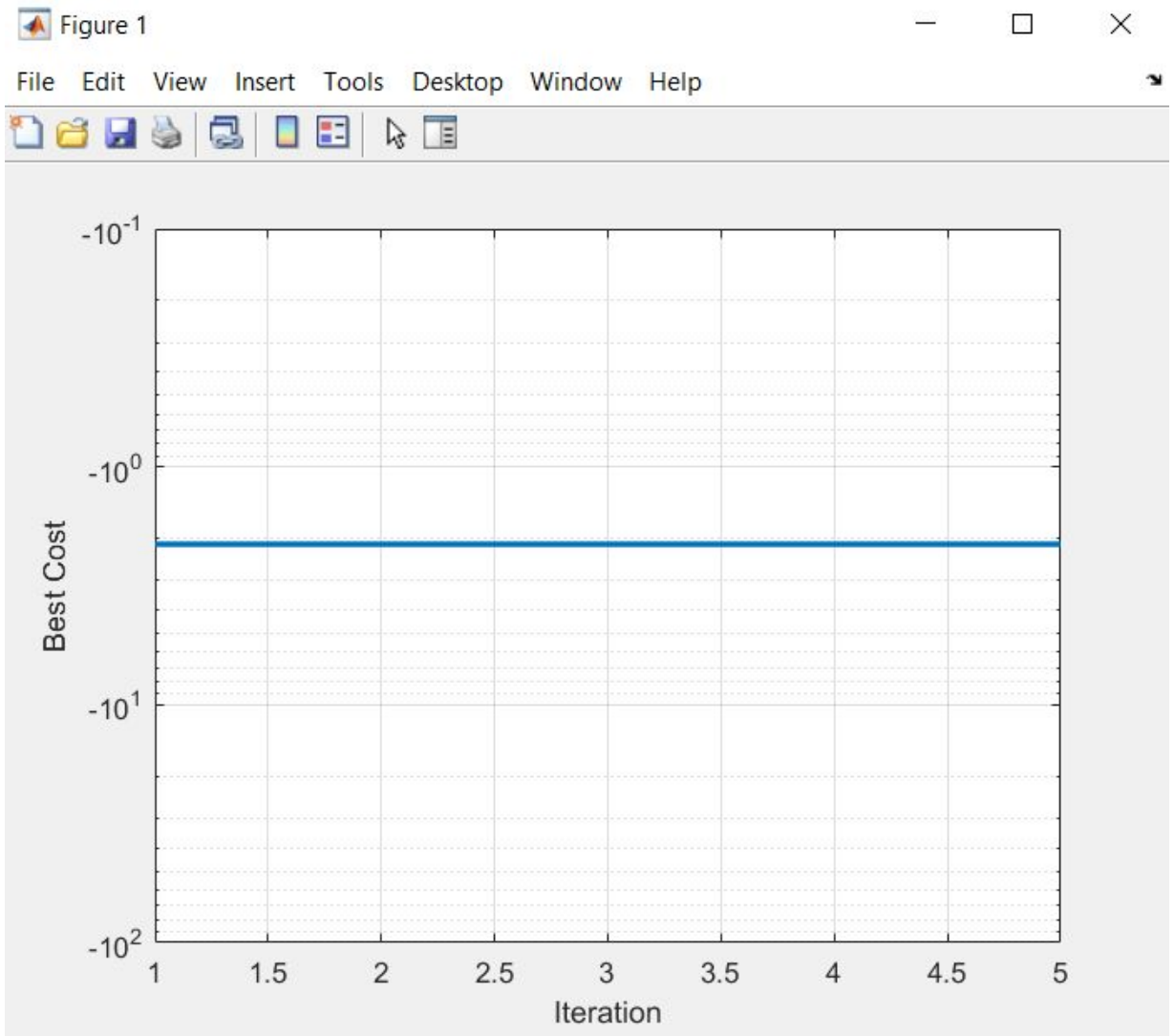
Watermark Recovered
from Bit Plane 3 of
Noise-Free Watermarked Image



Watermark Recovered
from Bit Plane 3 of
Noisy Watermarked Image



Cost function output graph for 5 iterations



MATLAB R2022a - academic use

HOME | PLOTS | APPS | EDITOR | PUBLISH | VIEW

New | Open | Save | Compare | Print | Go To | Find | Bookmark | Refactor | Profiler | Analyze | Run Section | Section Break | Run and Advance | Run to End | Run | Step | Stop

FILE | NAVIGATE | CODE | ANALYZE | SECTION | RUN

C: ▸ Users ▸ user ▸ OneDrive ▸ Desktop ▸ Mini Project ▸ review2

Current Folder

Name
- complexityFactor.m
- create_shares.m
- fireflyAlgo.m
- medianAttack.m
- noiseSaltPepper.m
- process.m
- share1.png
- share2.png

Details

Editor - C:\Users\user\OneDrive\Desktop\Mini Project\review2\fireflyAlgo.m

fireflyAlgo.m | process.m | complexityFactor.m | create_shares.m | medianAttack.m

```
54        pop(i).Cf = unifrnd(VarMin, VarMax, VarSize);
55        pop(i).Cost = process(pop(i).Cf);
56
57        if pop(i).Cost >= BestSol.Cost
58            BestSol = pop(i);
59        end
60    end
61
62    % Array to Hold Best Cost Values
63    BestCost = zeros(MaxIt, 1);
64
65    %% Firefly Algorithm Main Loop
66
67    for it = 1:MaxIt
68
69        newpop = repmat(firefly, nPop, 1);
70        for i = 1:nPop
71            newpop(i).Cost = inf;
```

Command Window

```
Iteration 1: Best Cost = -2.1139
Iteration 2: Best Cost = -2.1139
Iteration 3: Best Cost = -2.1139
Iteration 4: Best Cost = -2.1139
Iteration 5: Best Cost = -2.1139
fx >>
```

Workspace

Name	Value
alpha	0.1808
alpha_damp	0.9800
BestCost	[-2.1139;-2.1139;...
BestSol	1x1 struct
beta0	1
delta	0.0500
dmax	1
firefly	1x1 struct
gamma	1
i	20
it	5
j	20
m	2
MaxIt	5
newpop	20x1 struct
nPop	20
nVar	1
pop	20x1 struct
SortOrder	1x40 double
VarMax	1
VarMin	0
VarSize	[1,1]

Zoom: 100% | UTF-8 | CRLF | script | Ln 18 | Col 10

Type here to search | 26°C Partly cloudy | ENG US | 06:07 20/06/2022

Complexity factor and cost after 5 iterations for each firefly.

MATLAB R2022a - academic use

HOME | PLOTS | APPS | VARIABLE | VIEW

Search Documentation | Aditi

New from Selection | Open | Print | Rows | Columns | Insert | Delete | Transpose | Sort

VARIABLE | SELECTION | EDIT

C: ▸ Users ▸ user ▸ OneDrive ▸ Desktop ▸ Mini Project ▸ review2

Current Folder

Name
- complexityFactor.m
- create_shares.m
- fireflyAlgo.m
- medianAttack.m
- noiseSaltPepper.m
- process.m
- share1.png
- share2.png

Details

Editor - fireflyAlgo.m

Variables - pop

nPop | newpop | pop

20x1 struct with 2 fields

Fields	Cf	Cost
1	0.4982	-2.1139
2	0.2776	-2.1139
3	0.6525	-2.1139
4	0.9173	-2.1139
5	0.5098	-2.1139
6	0.9742	-2.1139
7	0.1973	-2.1139
8	0.1112	-2.1139
9	0.2974	-2.1139
10	0.3964	-2.1139
11	0.4208	-2.1139
12	0.3115	-2.1139
13	0.6938	-2.1139
14	0.0919	-2.1139
15	0.4021	-2.1139
16	0.2952	-2.1139
17	0.3065	-2.1139
18	0.1056	-2.1139
19	0.5938	-2.1139
20	0.2827	-2.1139
21		
22		
23		

Workspace

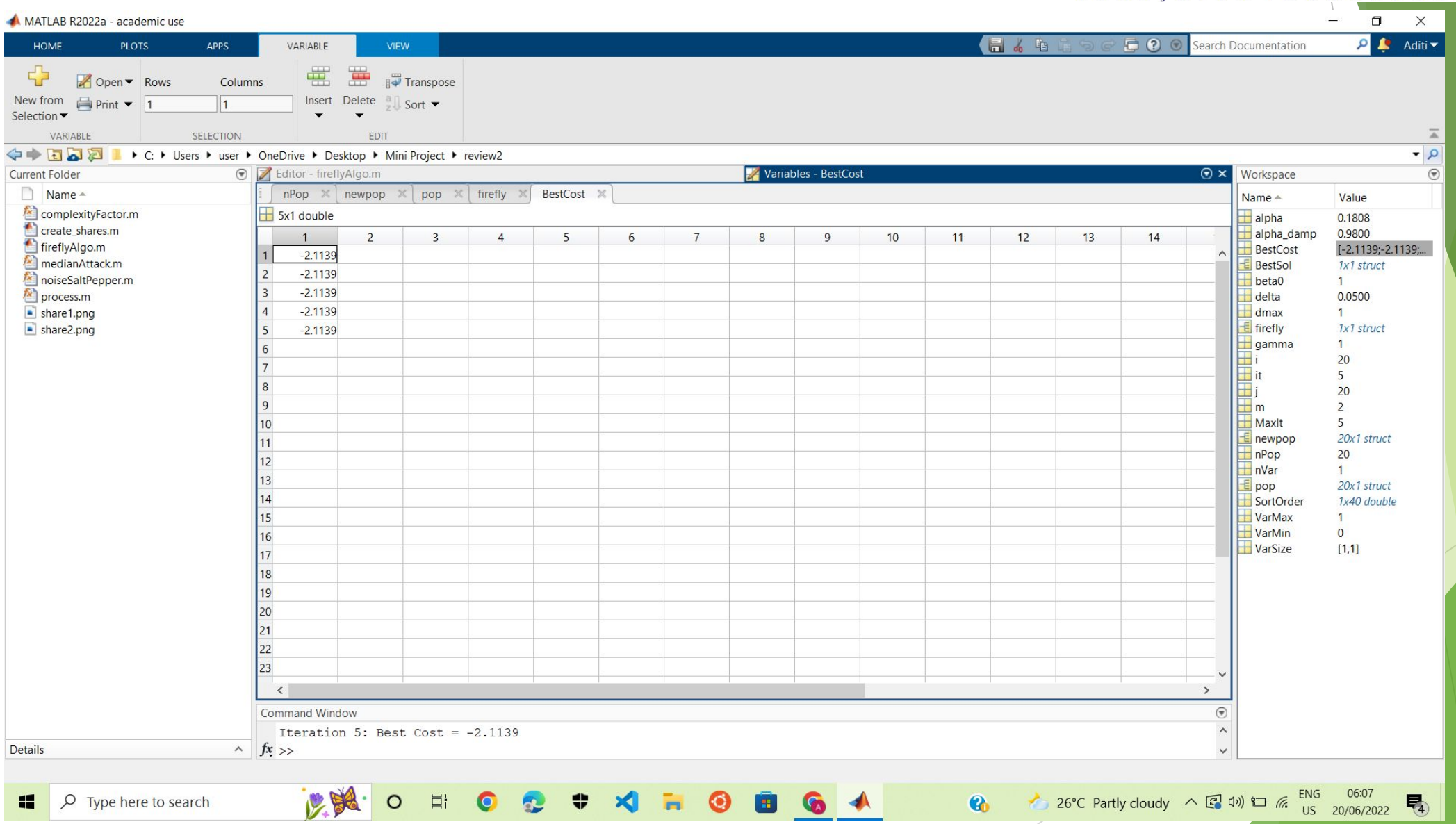
Name	Value
alpha	0.1808
alpha_damp	0.9800
BestCost	[-2.1139;-2.1139;...
BestSol	*1x1 struct*
beta0	1
delta	0.0500
dmax	1
firefly	*1x1 struct*
gamma	1
i	20
it	5
j	20
m	2
MaxIt	5
newpop	*20x1 struct*
nPop	20
nVar	1
pop	*20x1 struct*
SortOrder	*1x40 double*
VarMax	1
VarMin	0
VarSize	[1,1]

Command Window

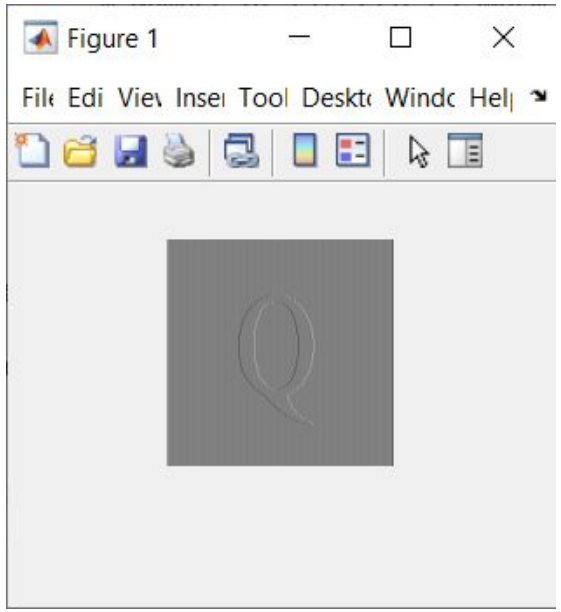
```
Iteration 5: Best Cost = -2.1139
>>
```

Type here to search | 26°C Partly cloudy | ENG US | 06:07 20/06/2022

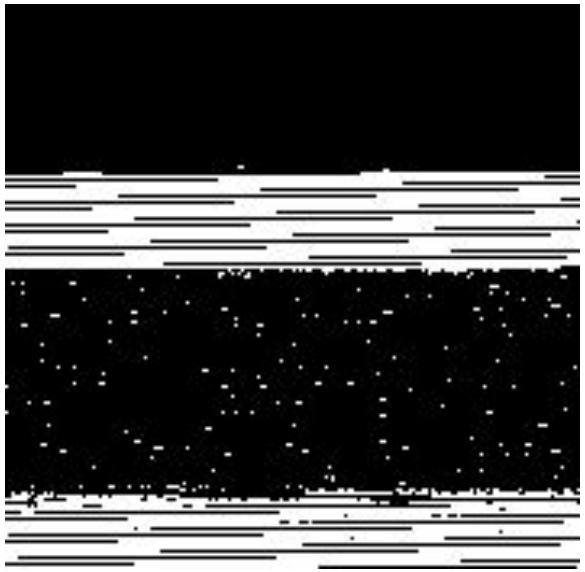
Best costs after each iterations



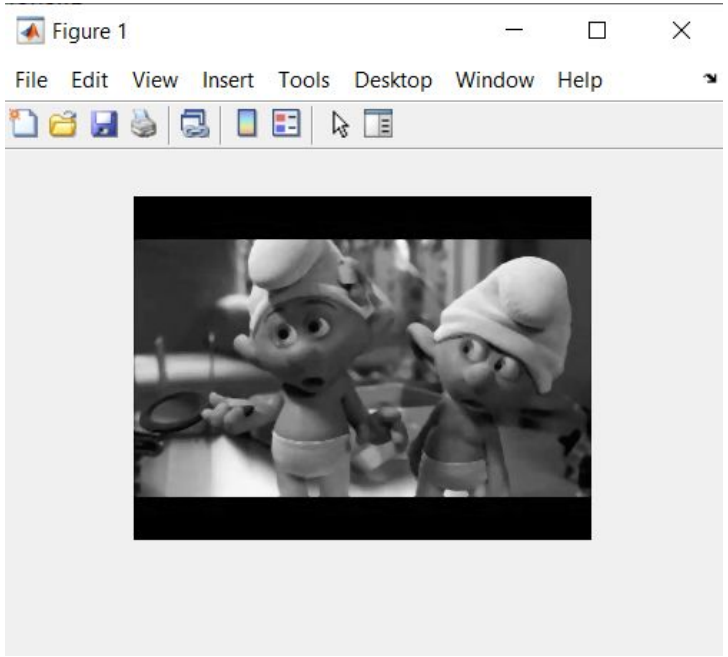
Embedded watermark



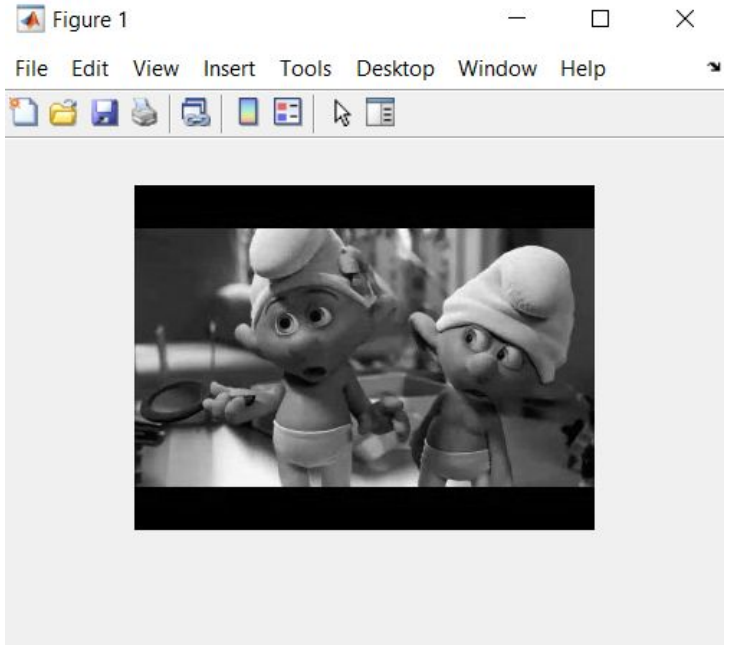
Extracted watermark



After embedding

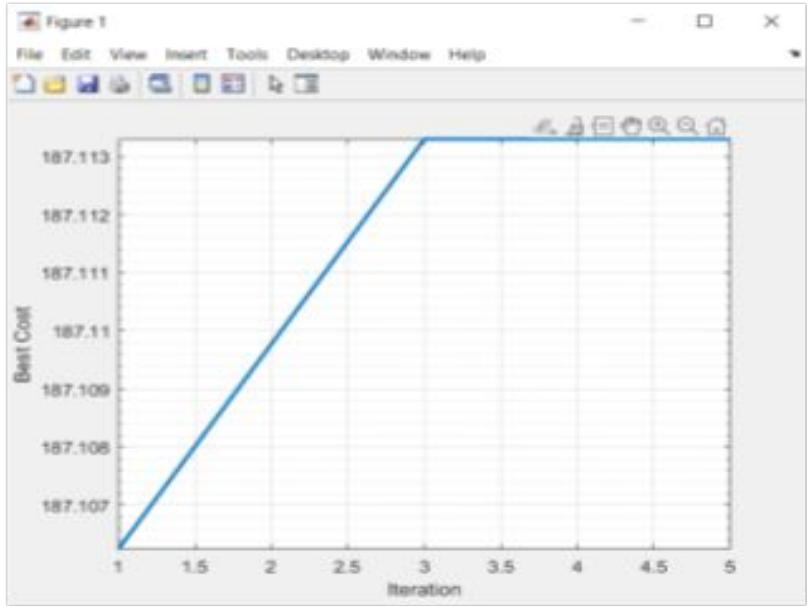


After extraction

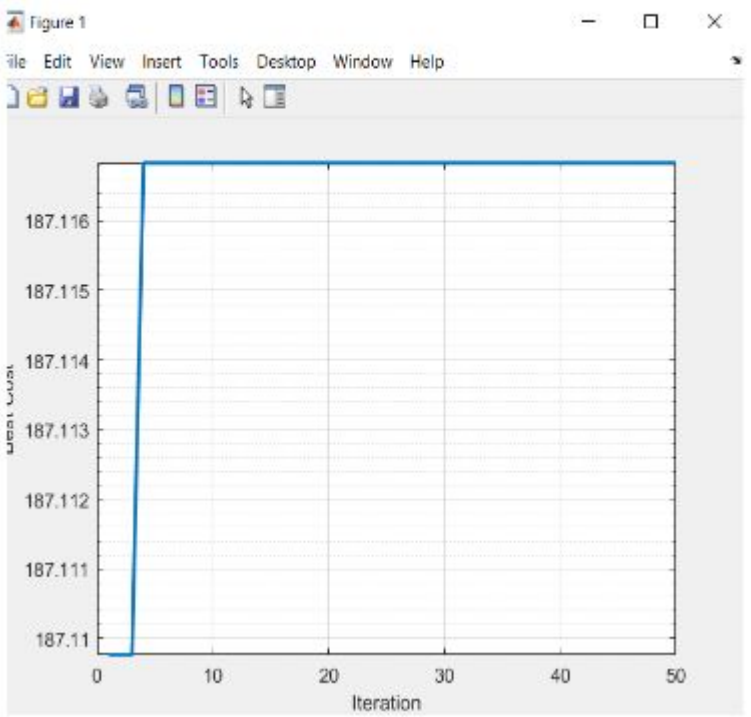


Results and discussion

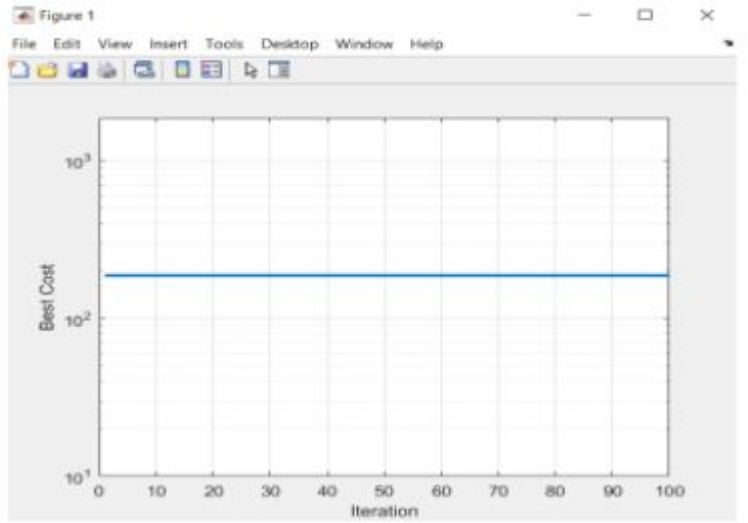
Video 1 with 25 iterations



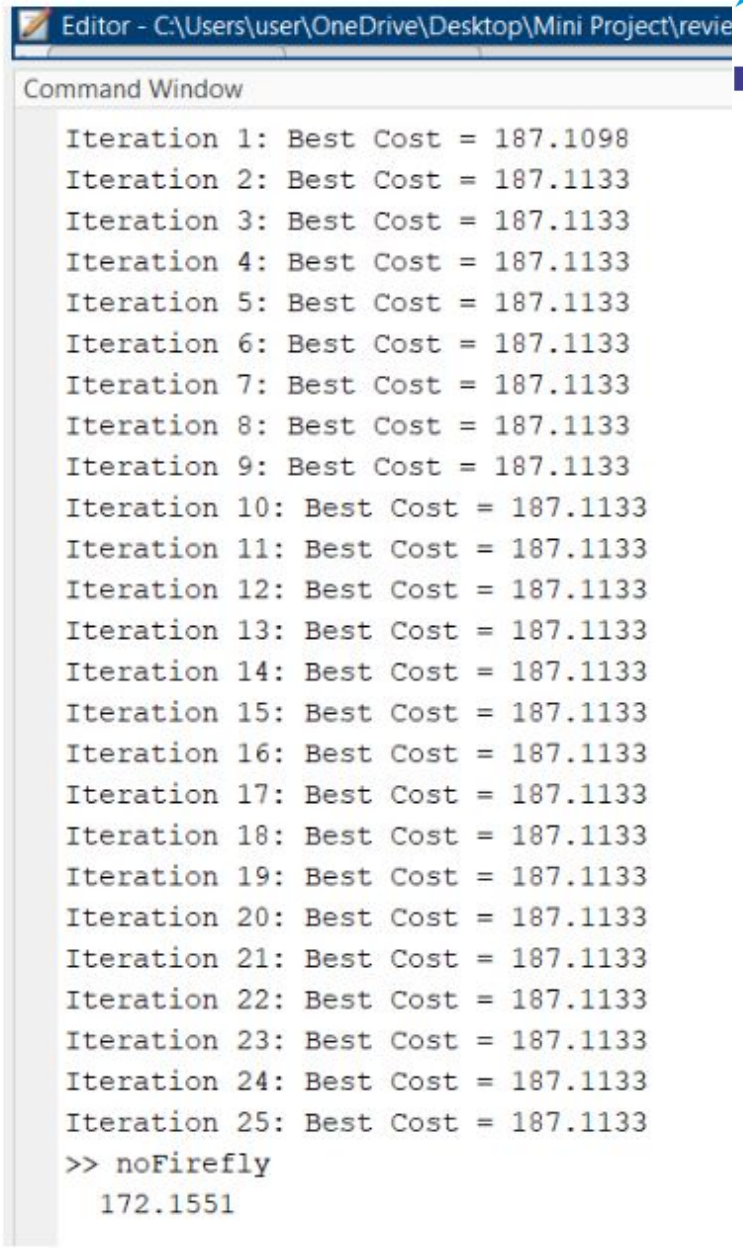
Video 1 with 50 iterations



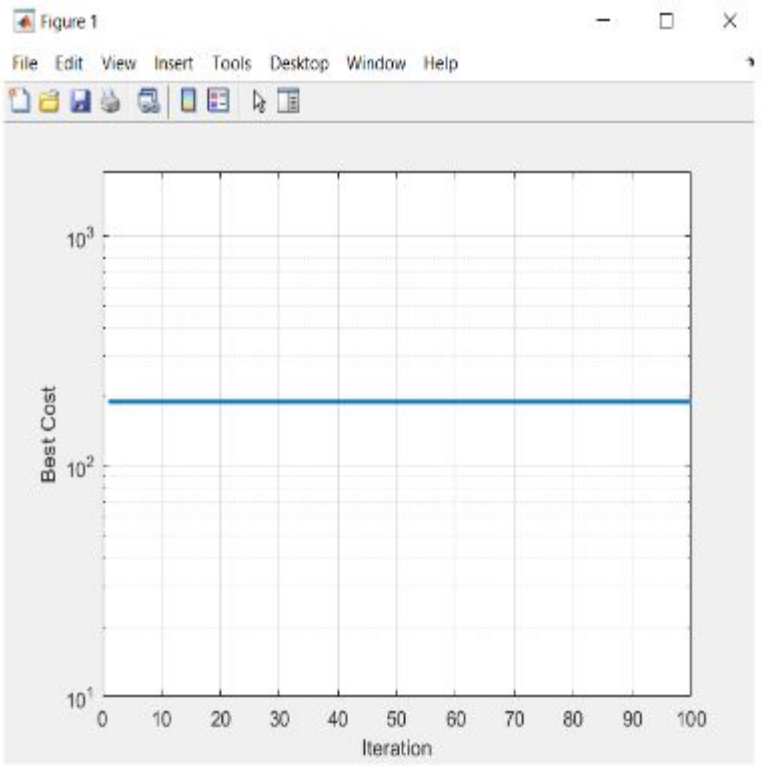
Video 1 with 100 iterations



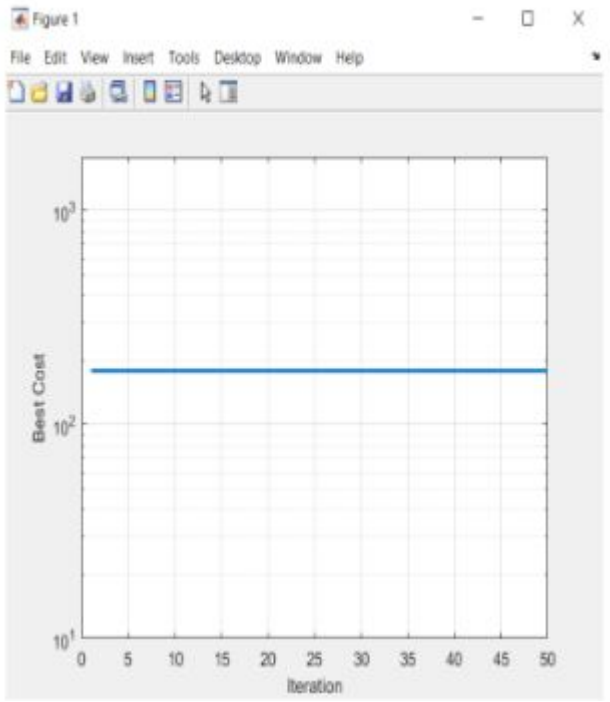
Video 1 with no firefly algorithm



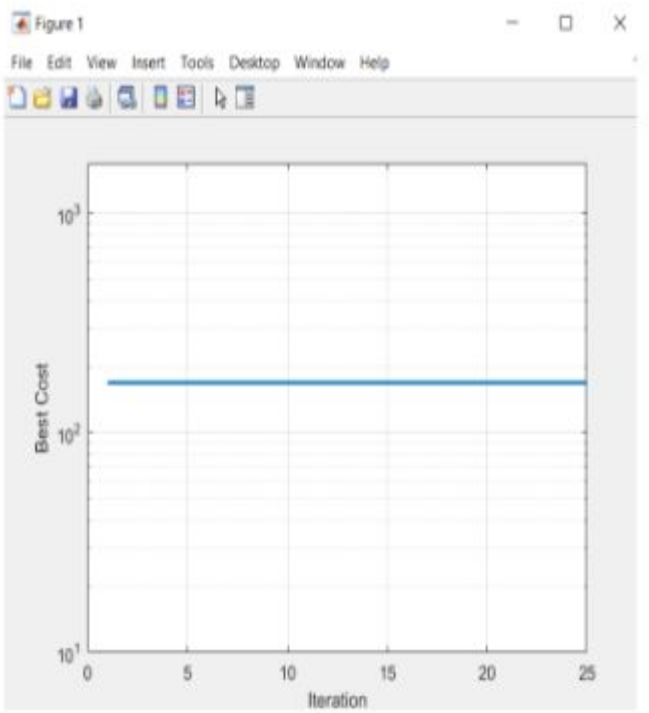
Video 2 with 100 iterations



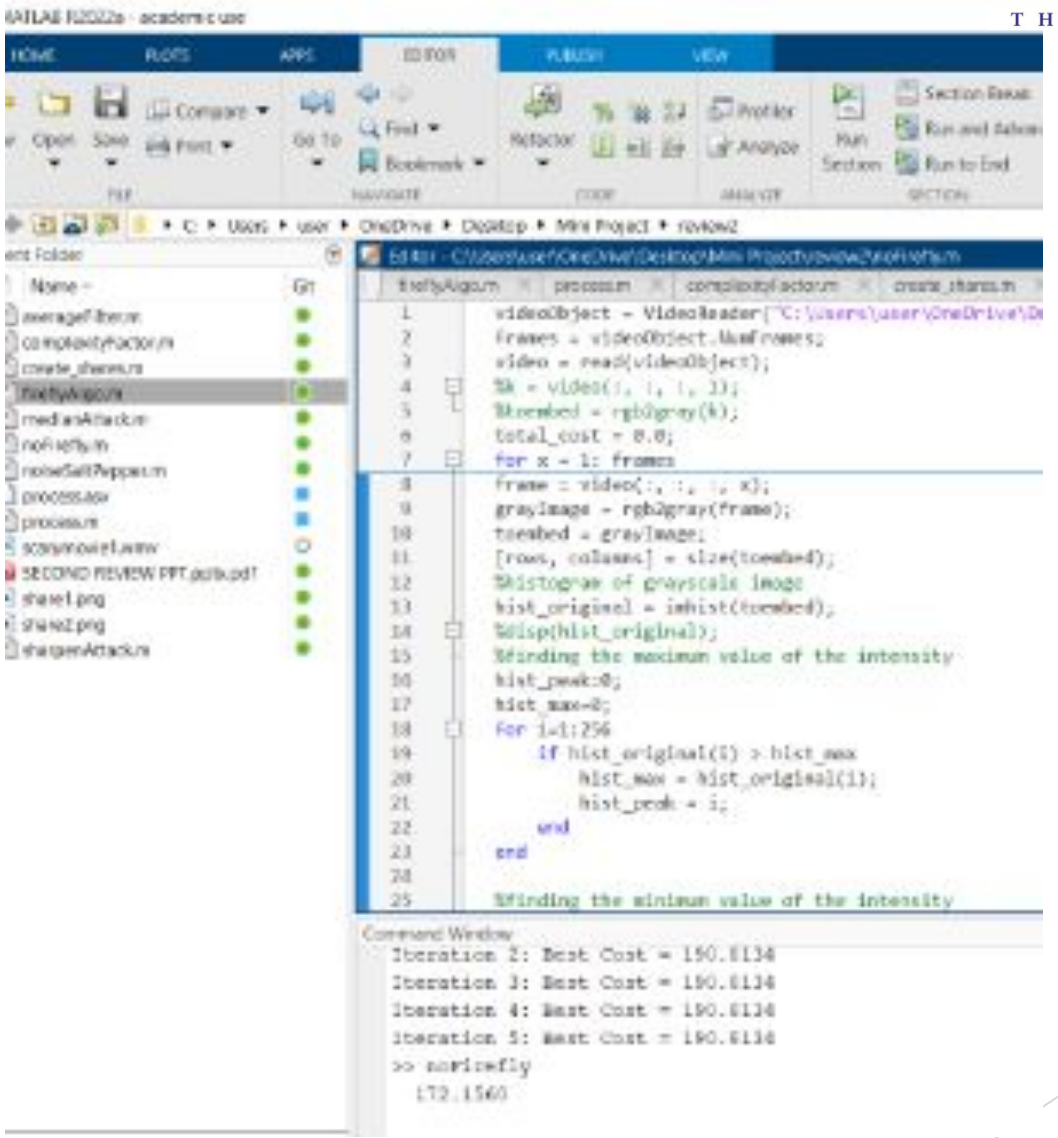
Video 2 with 50 iterations



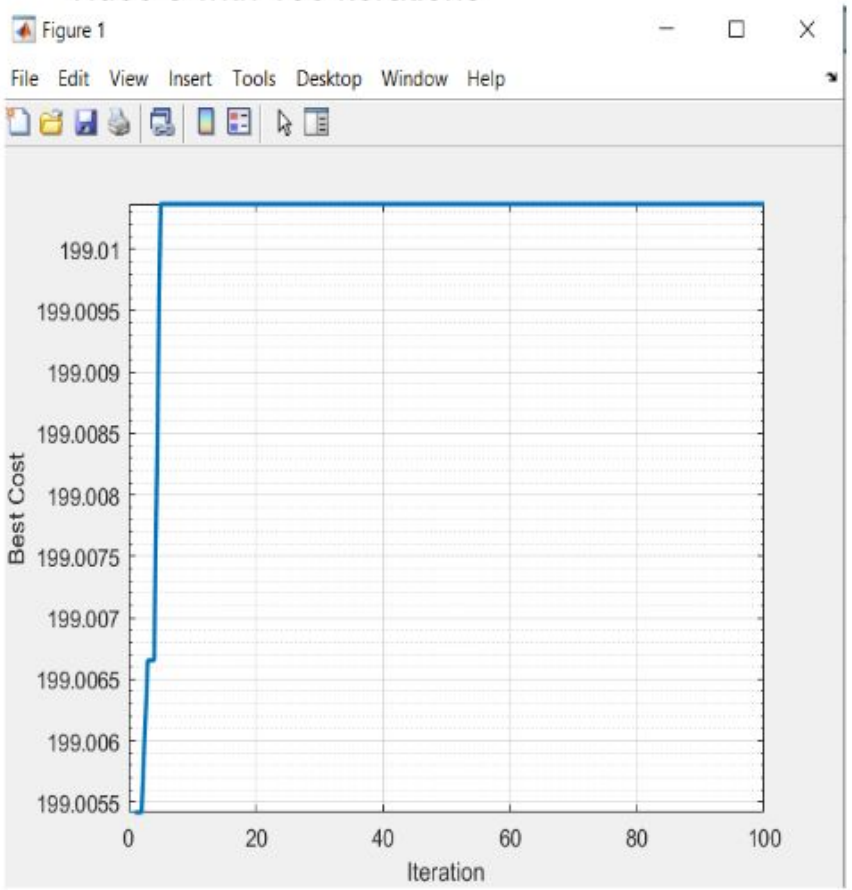
Video 2 with 25 iterations



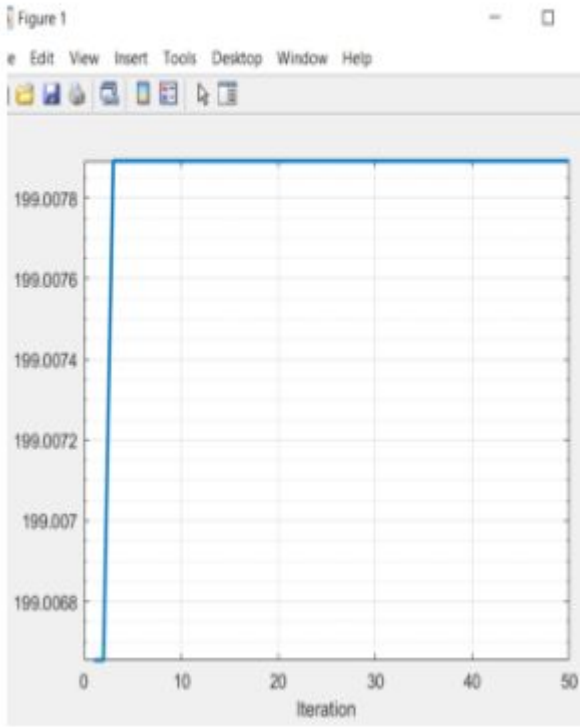
Video 2 with no firefly algorithm



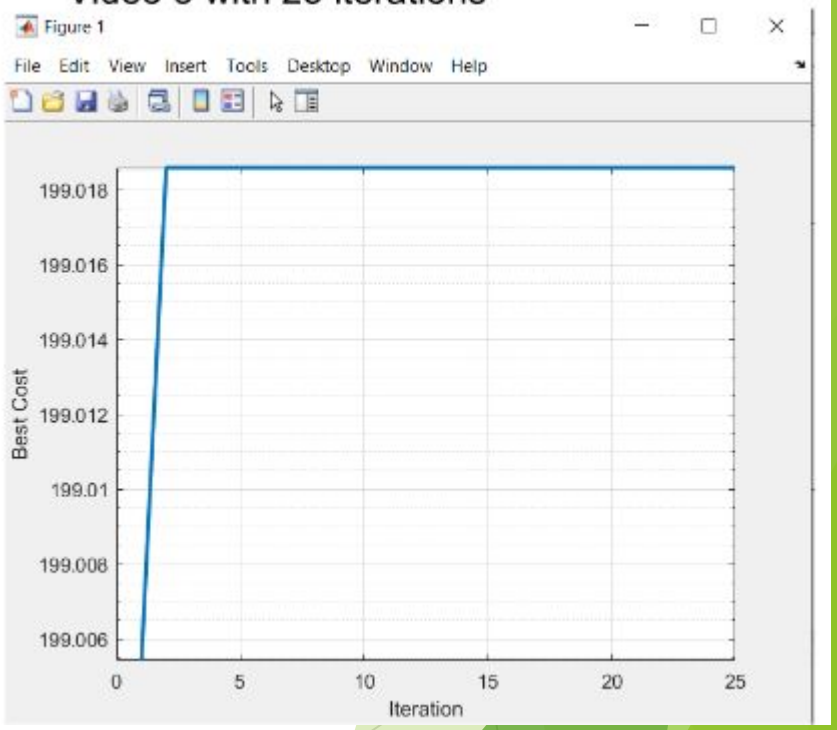
Video 3 with 100 iterations



Video 3 with 50 iterations



Video 3 with 25 iterations



Video 3 with no firefly algorithm

Editor - C:\Users\user\OneDrive\Desktop\Mini Project\review2\noFirefly.m

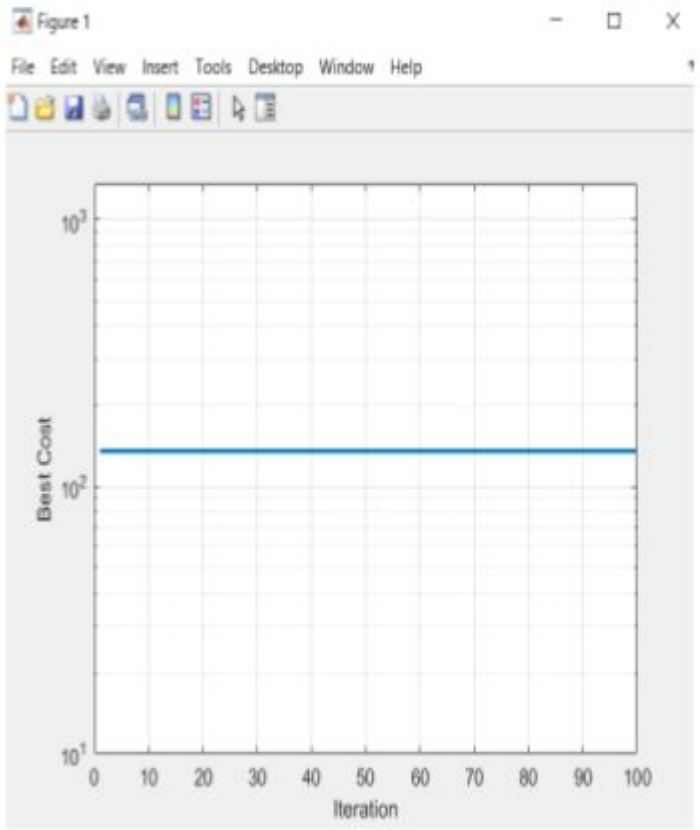
fireflyAlgo.m | process.m | complexityFactor.m | create_s

```
1   videoObject = VideoReader("iceage.wmv");
2   frames = videoObject.NumFrames;
3   video = read(videoObject);
4   %k = video(:, :, :, 1);
5   %toembed = rgb2gray(k);
6   total_cost = 0.0;
7   for x = 1: frames
8   frame = video(:, :, :, x);
9   grayImage = rgb2gray(frame);
10  toembed = grayImage;
11  [rows, columns] = size(toembed);
12  %histogram of grayscale image
13  hist_original = imhist(toembed);
14  %disp(hist_original);
15  %finding the maximum value of the intensity
16  hist_peak=0;
17  hist_max=0;
18  for i=1:256
19      if hist_original(i) > hist_max
20          hist_max = hist_original(i);
21          hist_peak = i;
22      end
23  end
```

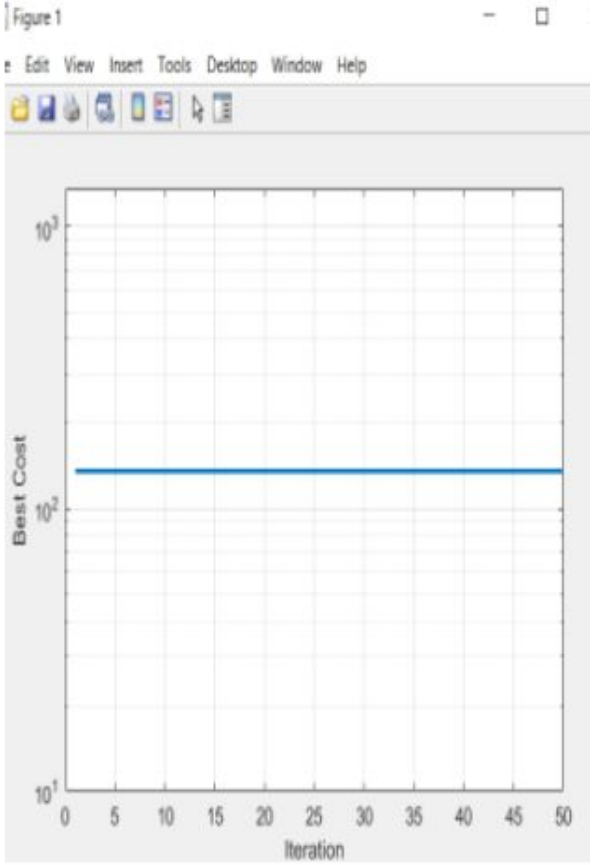
Command Window

```
>> noFirefly
   172.1558

>> noFirefly
   172.1552

>> noFirefly
   199.1721

fx >>
```

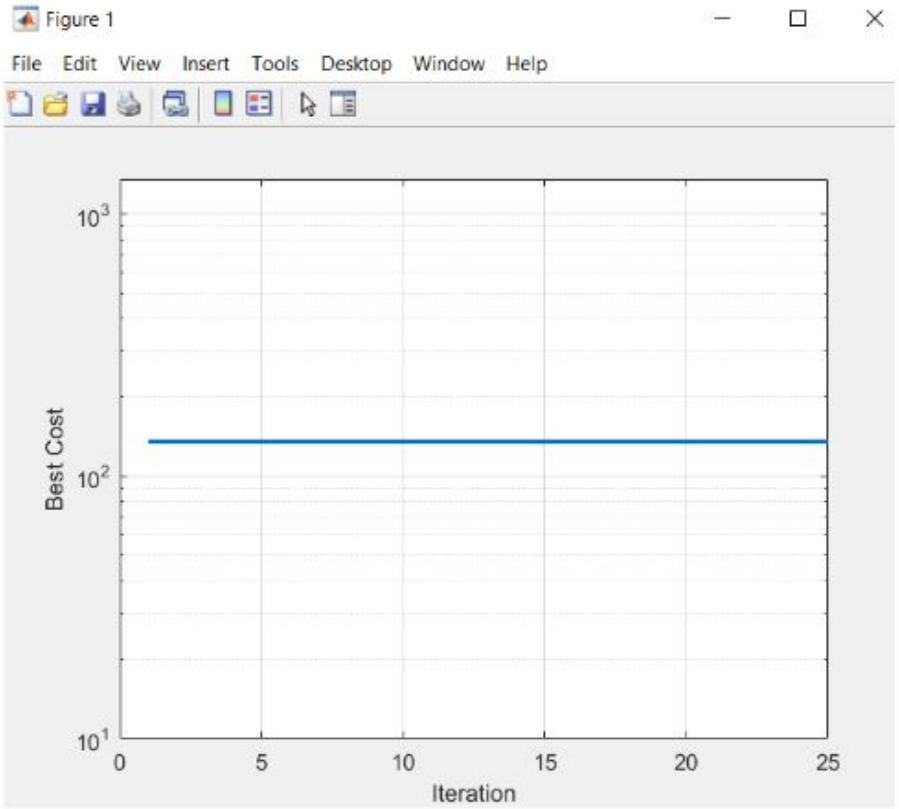
Video 4 with 100 iterations



Video 4 with 50 iterations



Video 4 with 25 iterations



Video 4 with no firefly algorithm

Editor - C:\Users\user\OneDrive\Desktop\Mini Project\review2\noFirefly.m

fireflyAlgo.m | process.m | complexityFactor.m | create_sha

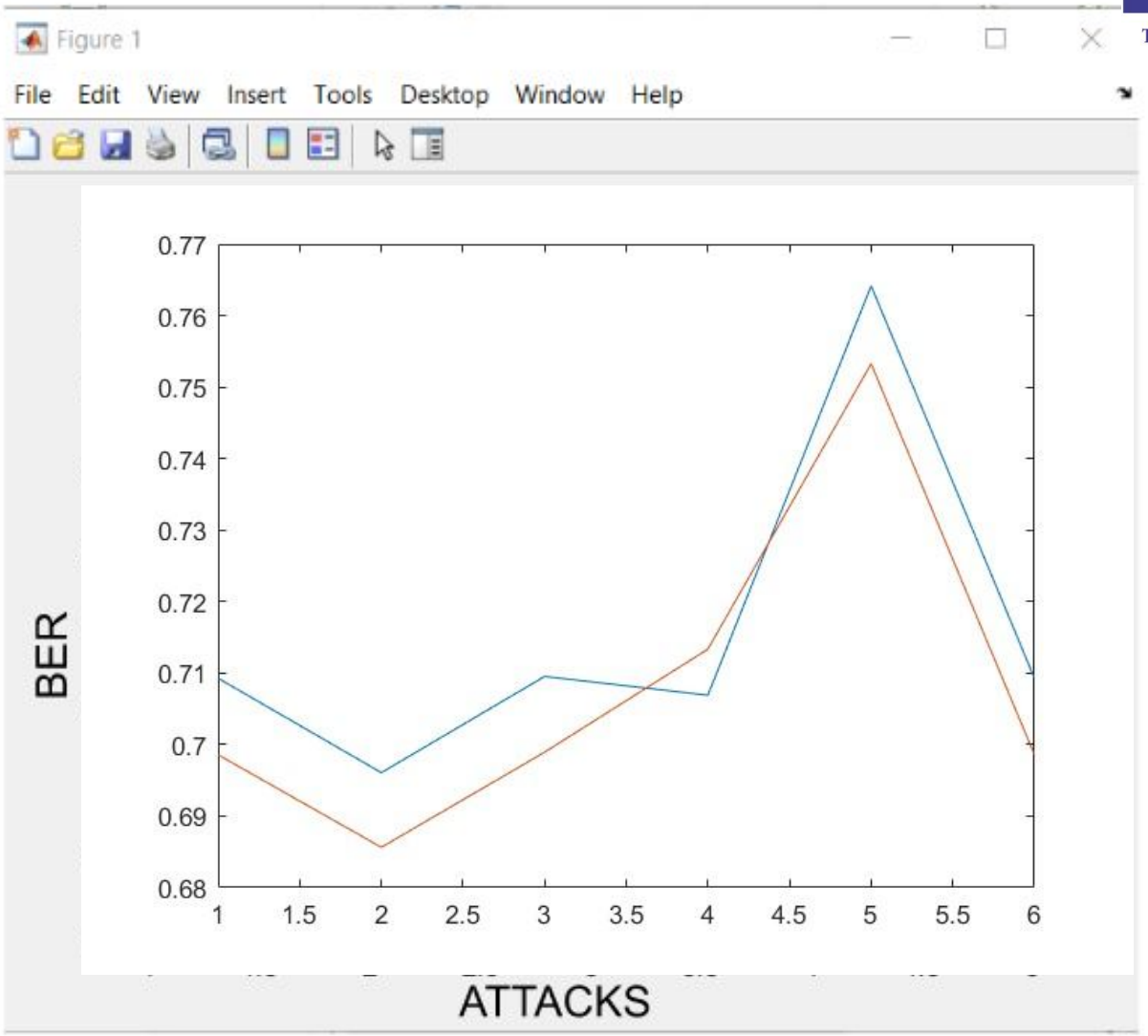
```
1   videoObject = VideoReader("smurfi2.wmv");
2   frames = videoObject.NumFrames;
3   video = read(videoObject);
4   %k = video(:, :, :, 1);
5   %toembed = rgb2gray(k);
6   total_cost = 0.0;
7   for x = 1: frames
8   frame = video(:, :, :, x);
9   grayImage = rgb2gray(frame);
10  toembed = grayImage;
11  [rows, columns] = size(toembed);
12  %histogram of grayscale image
13  hist_original = imhist(toembed);
14  %disp(hist_original);
15  %finding the maximum value of the intensity
16  hist_peak=0;
17  hist_max=0;
18  for i=1:256
19      if hist_original(i) > hist_max
20          hist_max = hist_original(i);
21          hist_peak = i;
22      end
```

Command Window

```
Iteration 95: Best Cost = 135.9089
Iteration 96: Best Cost = 135.9089
Iteration 97: Best Cost = 135.9089
Iteration 98: Best Cost = 135.9089
Iteration 99: Best Cost = 135.9089
Iteration 100: Best Cost = 135.9089
>> noFirefly
  138.5073

fx >>
```

With firefly Without firefly

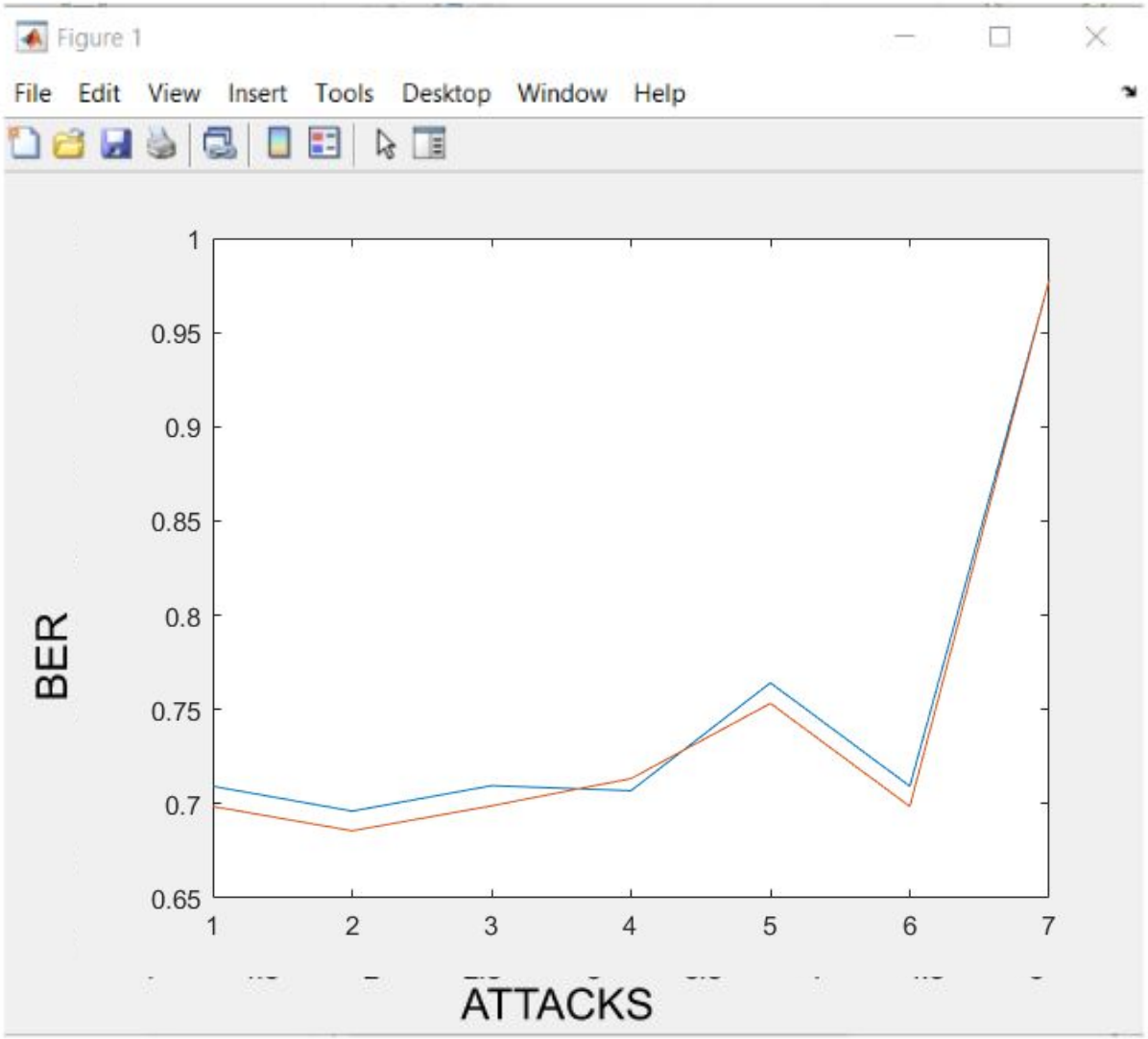
1 - no attacks
2 - median attack
3 - salt and pepper noise
4 - average filter
5 - sharpen attack
6 - speckle noise



With firefly

Without firefly

1 - no attacks
2 - median attack
3 - salt and pepper noise
4 - average filter
5 - sharpen attack
6 - speckle noise
7 - gaussian noise



Merits and Demerits of the work

MERITS

- Histogram bit shifting is used, which gives a good data hiding capacity for reversible data hiding.
- Gives low BER when firefly algorithm is used, than the usual method.

DEMERITS

- The firefly algorithm has probability of being trapped in local optima because they are local search algorithms.
- Processing time is higher because the threshold value changes for each video.

Conclusion and future work

In this project first we create two shares using visual cryptography in which one is used for embedding and other is used later while making the whole watermarked video. For embedding we find a threshold value through firefly algorithm and embed it to those frames using histogram bit shifting which is a reversible data hiding technique. And we perform a few attacks as explained in our project and finally extract the watermarked image and see for any distortions. We can see an increase in PSNR values which is good as it means it has good quality.

Hence, in the future, this technique can be tested for other geometrical attacks and we can also improve the technique for embedding watermarks in the video.

REFERENCES

1. **Amirtharajan, R. & Sulthana, Samreen & Rayappan, John Bosco Balaguru,** (2013). Seeing and Believing is a Threat: A Visual Cryptography Schemes. Research Journal of Information Technology. 5. 435-441. 10.3923/rjit.2013.435.441.

2. **Surafel Luleseged Tilahun, Hong Choon Ong,** (2012) "Modified Firefly Algorithm", Journal of Applied Mathematics, vol. 2012, Article ID 467631, 12 pages, 2012.

3. **Ali, Musrrat & Ahn, Chang Wook & Pant, Millie.** (2016). Intelligent Watermarking Scheme Employing the Concepts of Block Based Singular Value Decomposition and Firefly Algorithm. 10.15579/gcsr.vol5.ch3.

4. **K. Vijaya Durga; G. Mamatha; Ch. Hima Bindu** (2015). SVD based image watermarking with firefly algorithm.

5. **Huailin Dong; Mingyuan He; Ming Qiu** (2015). Optimized Gray-Scale Image Watermarking Algorithm Based on DWTDCT-SVD and Chaotic Firefly Algorithm.

6. **Huang H-C, Chen Y-H, Abraham A** (2010) Optimized watermarking using swarm-based bacterial foraging. J Inf Hiding Multimed Signal Process 1(1):51–58

7. Firefly algorithm udemy - https://www.udemy.com/course/firefly-optimization-algorithm-in-matlab/learn/lecture/11336#reviews
8. Firefly algorithm code github - https://github.com/smkalami/ypea112-firefly-algorithm/tree/master/Firefly%20Algorithm

9. **Kumar, M., Aggarwal, J., Rani, A.** *et al.*(2022). Secure video communication using firefly optimization and visual cryptography. *Artif Intell Rev* 55, 2997–3017.

THANK YOU !