Detecting Computer Generated Images

Team Members : Joydeep Das
Ravi Chandora
Swabhiman Patnaik

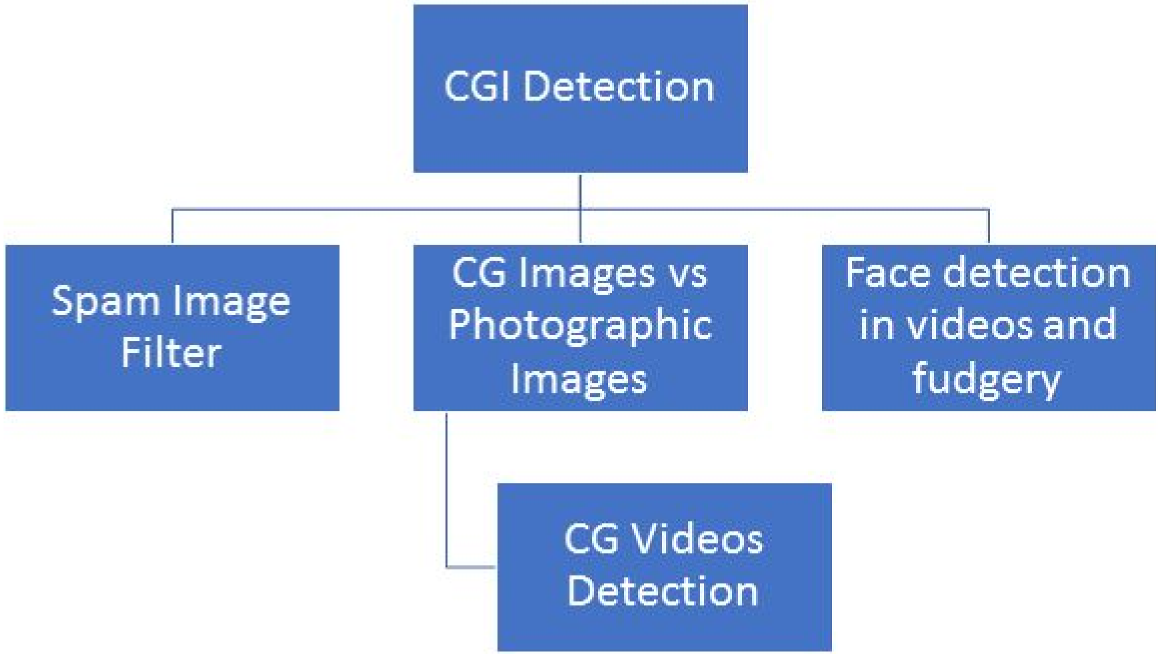
Motivation

- With advancements made in the field of computer technology, generating images with the help of softwares has become quite common.
- This technology is used for several productive fields, such as in printed media, video games, films, television programs, shorts, commercials, videos, and simulators. As a result, 1990s saw the birth of virtual cinematography wherein laws of physics could be defied.
- On the other hand, CG technology can be used by vicious groups to forge false & deceiving images or videos. Due to the advancement in computer graphics come to appear so photo-realistic that it may be used as a convincing form of photographic image forgery.

Introduction

- Over the course of this project we have implemented three papers, the first two dealing with still images and the third paper is video-based.
  - The first method helps in **Spam Image Filtering** based on two features and SVM machine learning.
  - The second paper revolves around the **Differentiation of photorealistic CG images and photographic images**.
  - The third paper provides a method to analyse **fudgery in videos containing human faces**.

Methods Implemented


CGI Detection
Spam Image Filter
CG Images vs Photographic Images
Face detection in videos and fudgery
CG Videos Detection

Spam Image Filtering

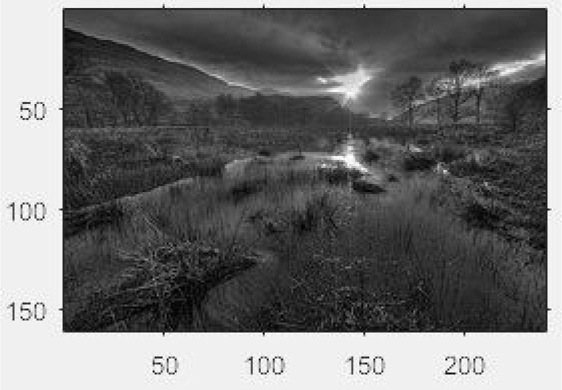
Filtering CG images used for Image spamming

- ❖ These spams are basically images embedded with text messages.
- ❖ For spam images, the background should have high contrast to the text part for users to read the text message. Mostly, a light (white) background is preferred, thus exploring the lightness of HSL.

Implementation

❖ First, the image is obtained in RGB domain. It is converted to HSV domain using the following equations:-

→

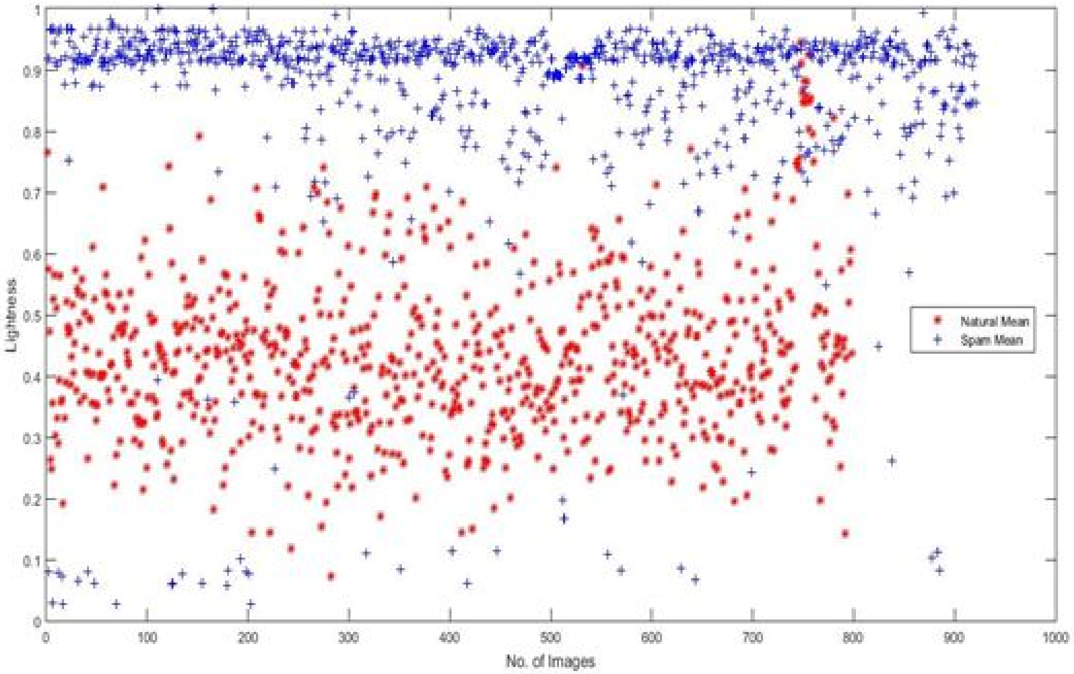


RGB Image

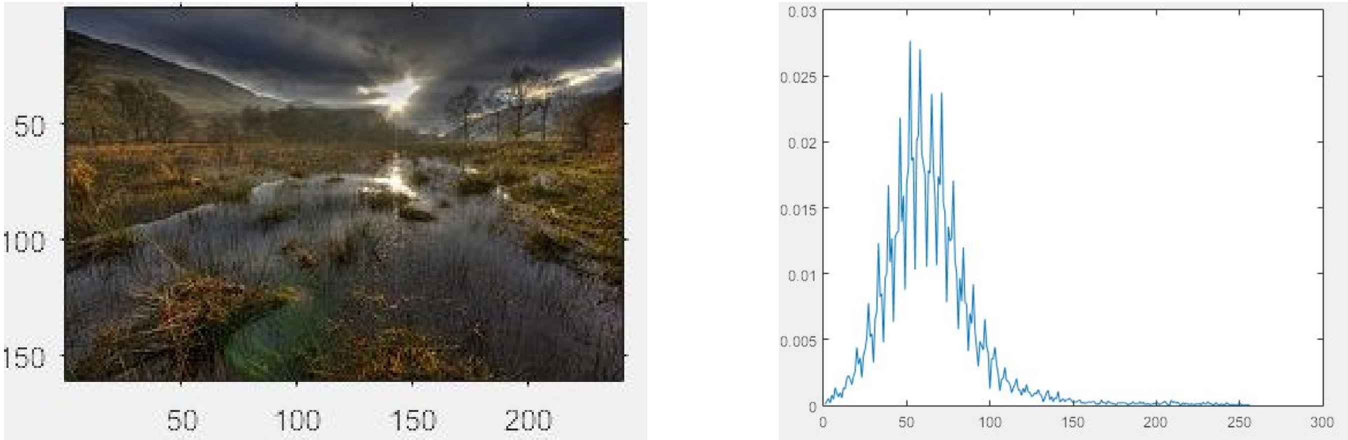
HSL Image (Lightness component)

❖

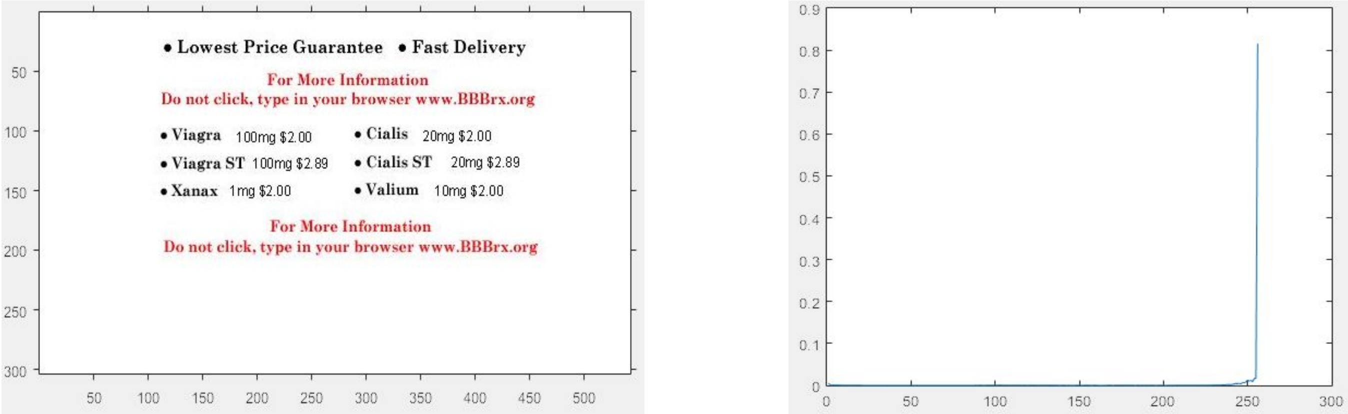
- Lightness Domain→ Average Lightness or Mean Lightness.
- This was applied on the dataset of 921 Spam Images and 798 Photographic images. The values were plotted on a graph:-



(a)Lightness Values of dataset



(a). Natural image and its lightness histogram



(a). Spam image and its lightness histogram

- ❖ Clearly for spam image above lightness value is above 0.5, and natural image spread over 0.2 to 0.7.
- ❖ For the second feature, rate of change, we use the following formula:-

  rate of change = $| (maxa - maxb) / maxb | \times 100$.

  maxa=max. histogram value before removal of all histogram bins above threshold = 0.05.

  maxb= max. historam after removal.

  If rate is more than 10% then peak is observed, declared Spam Image

Results

1. Unsupervised method (or Direct Method)

   True Positive rate = **93.27%** False Negative rate = **6.73%**

   True Negative rate = **92.61%** False Positive rate = **7.39%**

2. Supervised method (using SVM trainer using the two features)

   True Positive rate = **97.10%** False Negative rate = **2.90%**

   True Negative rate = **91.35%** False Positive rate = **8.65%**

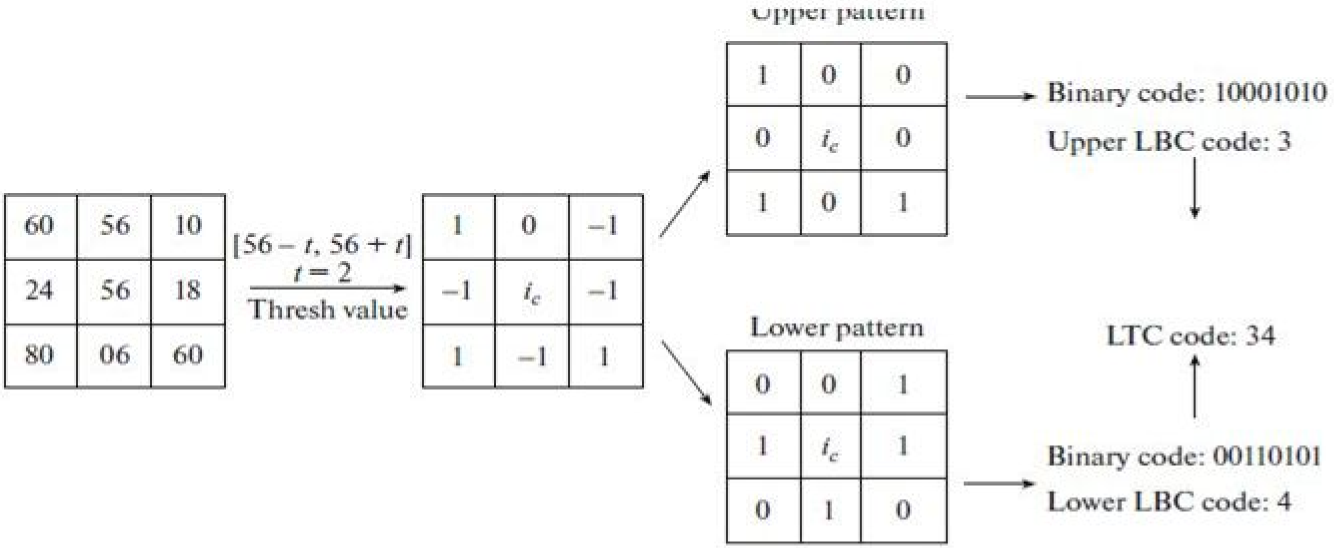
General CGI vs Natural Images

Distinguishing photorealistic CG images from natural images

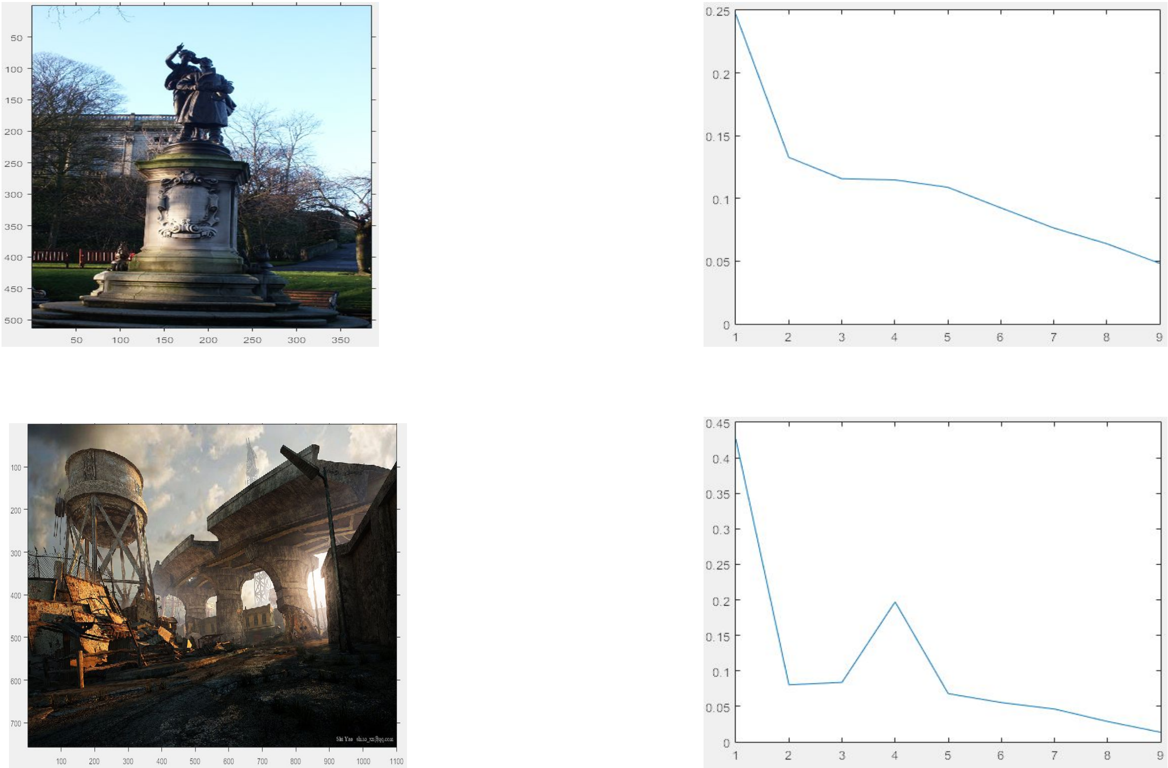
- We Exploit the fact that PIM changes slower in brightness, colour, texture and have richer texture level than PRCG.
- Based on less feature dimensions, LTC feature extracted ,tested on SVM and serves as good classifier between PIM and PRCG.

Implementation

- ❖ LTC feature extraction: Basic goal is to extract smoothness information in three individual HSV plane thereby getting idea about color, texture and brightness.



- Upper LBC and lower LBC both can have [0,8] i.e 9 values each, plot histogram based on entire image LBC’s. For e.g. -



(a). Natural image and its histogram (b) CGI Images and its histogram

Results

SVM accuracy :-

True Positive rate	= **95.26%**	False Negative rate	= **4.74%**
True Negative rate	= **99.54%**	False Positive rate	= **0.46%**

CG elements in Video

Extension of the previous paper

- We took the implementation of the previous paper and extended the algorithm to videos.
- We are doing a frame by frame analysis to detect software developed components in a video.

Procedure

- ❖ From the video, we extract the rich-texture frames. (Rich texture frames are the frames with maximum standard deviation.)
- ❖ Videos => Frames => Rich-texture frames => Feature extraction => Classification
- ❖ The trainer was trained with some sample videos and then it was tested on few frames of a test video.

Results

The accuracy obtained in this method:-

True Positive rate	= **90.00%**	False Negative rate	= **10.00%**
True Negative rate	= **87.00%**	False Positive rate	= **13.00%**

Detection of CG Faces in Videos

Based on Human pulse signal detection

- This method exploits a naturally and universally common signal to humans, pulse signal.
- A relatively stable facial landmark is chosen to extract the pulse signal, i.e. the nose or forehead.

How to detect human pulse signal?

- Start with an image number i and Cth pixel where C belongs to{R,G,B}.

$$C_i = I_{Ci}(\rho_{Cdc} + \rho_{Ci})$$

Where Ici= Average light source illumination over which Ci is captured, Pcdc is the dc refefelection coeeficient known to us and pci is time varying ac coeffient due to pulsation of blood vessels.

- To detect pulse signal only and removing other variations:
  - To reduce effect of chaning light illumination Ici: Take normalise Ci over a window span including at least a pulse.

$$C_{ni} = \frac{C_i}{\mu(C_i)}$$

Reduce Motion effects

- Consider light source to be constant now, and skin moves w.r.to it.
  - Over {R,G,B} the illuminance is constant, but pulsation of blood volumeis different in different domain. Ratio will equalize them and hence, G over B is taken as its shows maximum robustness.

$$S_i = \frac{G_{ni}}{R_{ni}} - 1$$

- Recent research shows one more component to be handled, which is Pc the term which doesnot penetrates and passes straight after reflection.

$$C_i = I_{Ci}(\rho_{Cdc} + \rho_{Ci} + s_i)$$

Where Si shows for white light illumination equal distribution and thus taking perpendicular X=R-G and Y=0.5*R+0.5*G-B component.

Final pulse signal

- Combining for motion and specular reflection coefficient : a pulse signal

$$S = \frac{X_n}{Y_n} - 1$$

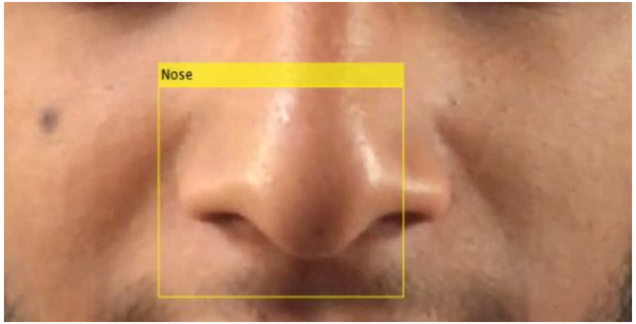
Where Xn and Yn are defined for any colour illuminance by considering normalised ratio [Rs,Gs,Bs]=[0.7682, 0.5121, 0.3841].

- Accordingly Xn=3*Rn-2*Gn and Yn= 1.5*Rn+Gn-1.5*Bn.

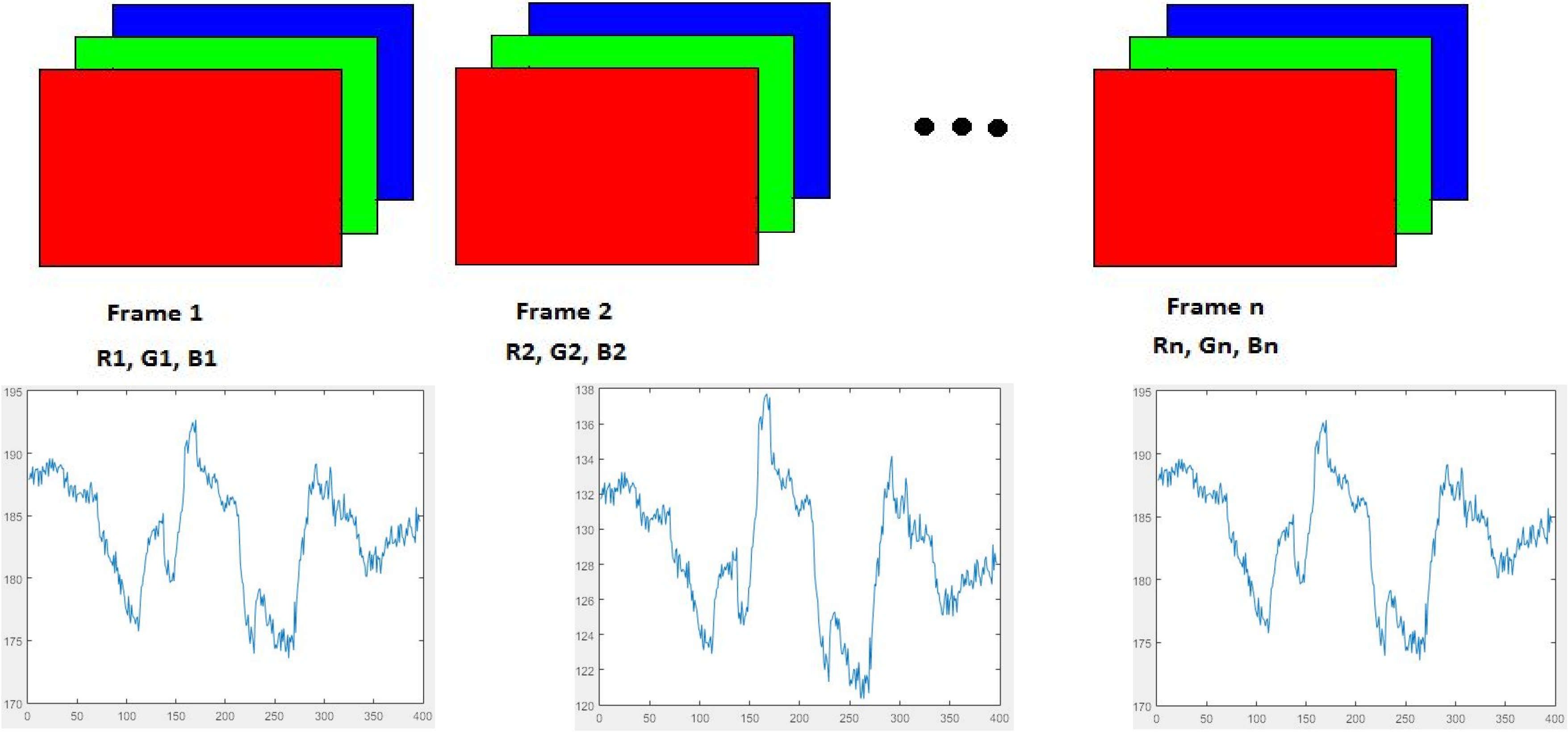
- Pulse Signal is then analysed over human pulse rate spectrum of 50~80bpm , to find a peak signal around that frequency.

Implementation

- ❖ Firstly, we detect the face in the video under analysis, frame by frame. Then, the nose area is extracted from the detected face.



❖ Here, Ri, Gi, Bi are the mean of all R, G, and B values of that frame.



Mean R, G, B over frames

- Let Cn be the normalized version of C where C ∈ {R,G,B}

$$C_{ni} = \frac{\sum_{j=-n}^{n} C_{i+j}}{2n+1}$$

- The RGB matrixes for the extracted region are taken and X,Y parameters are calculated:-

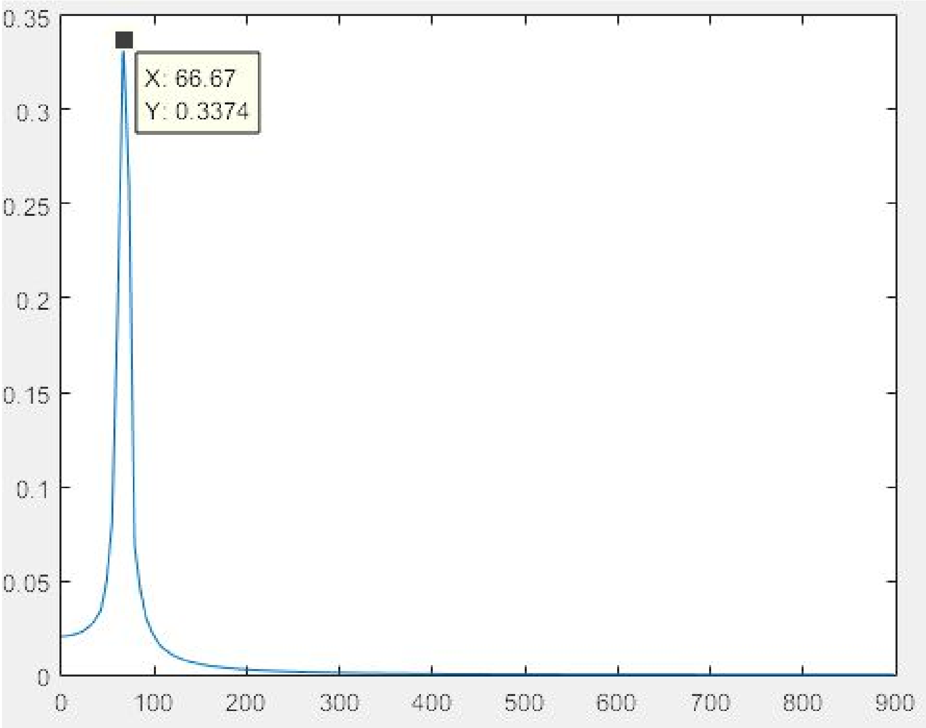
$$X = 3R_n - 2G_n$$
$$Y = 1.5R_n + G_n - 1.5B_n$$

- To eliminate the disturbance from irrelevant frequencies, X, Y is bandpass filtered to get Xf, Yf . Then the pulse signal is obtained as

$$S = X_f - \alpha Y_f$$

- where, α is the ratio of the standard deviations of X_f and Y_f

$$\alpha = \frac{\sigma(X_f)}{\sigma(Y_f)}$$

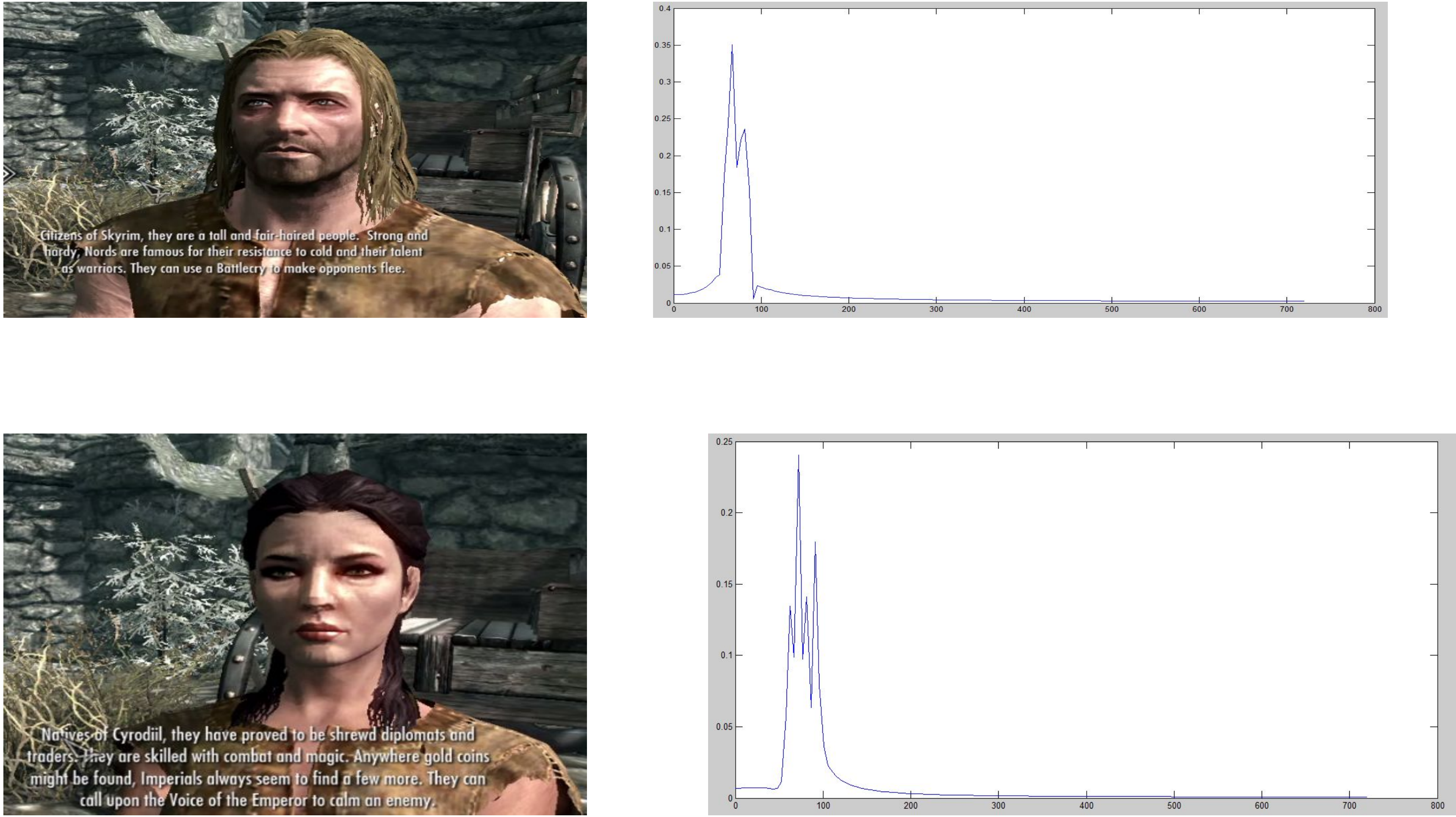


A pulse signal is obtained after passing the signal through passband of frequency 50 ~ 80 bpm (the natural human pulse rate lies in this range). Then, DFT was applied to the extracted pulse signal to obtain the frequency spectrum like the above figure.

Results

For forged videos:-

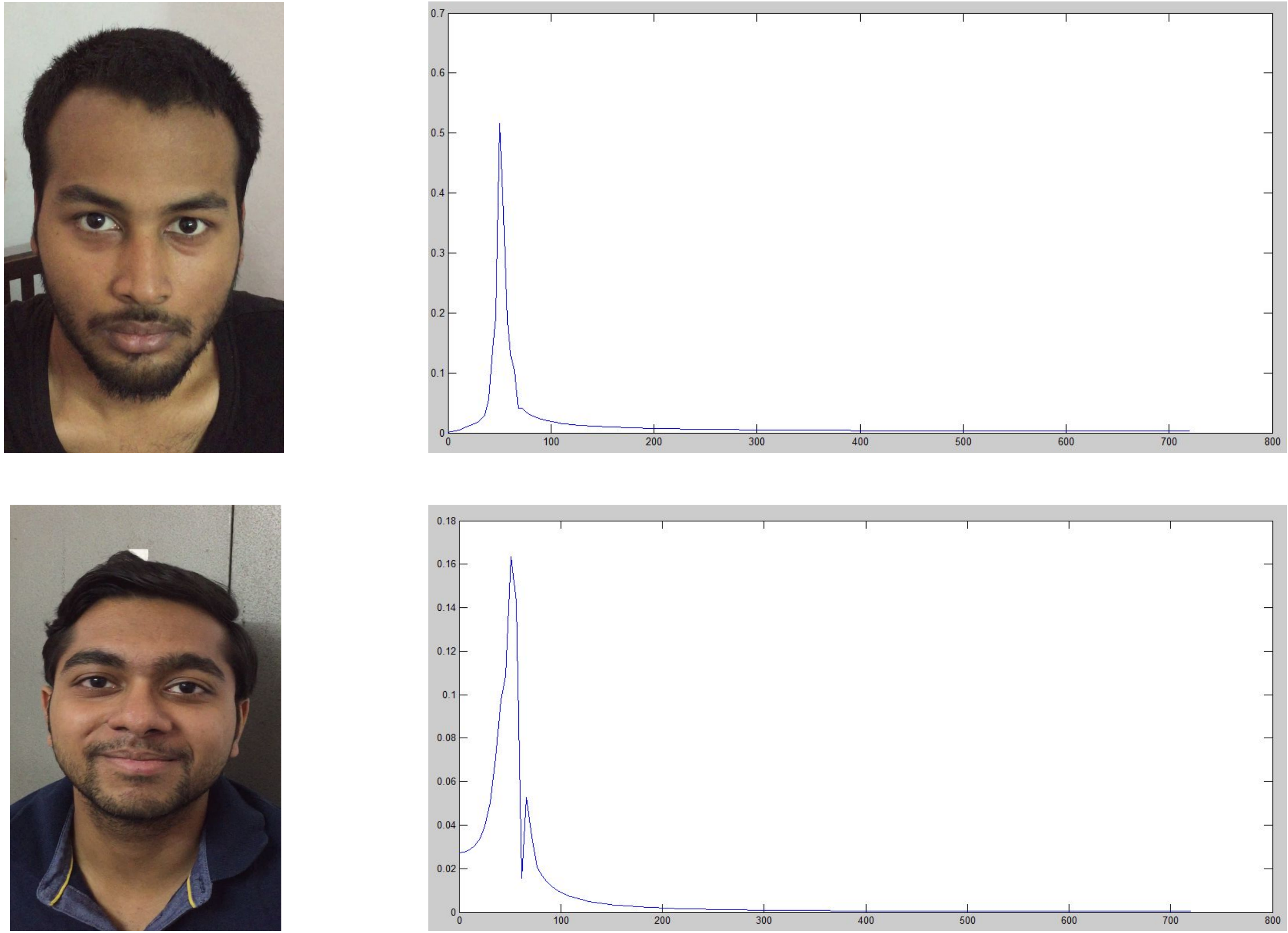
Serial no.	Frame rate	1st Highest peak (x)	2nd Highest peak (y)	(x-y)/y
1	21.2	0.0235	0.011	1.091
2	21.2	0.35	0.24	0.45
3	20.9	0.36	0.22	0.63
4	19.8	0.21	0.115	0.82
5	19.9	0.32	0.26	0.23
6	20.8	0.245	0.215	0.319
7	20.6	0.27	0.19	0.42



(a). CGI Face video and its Pulse

For genuine videos:-

Serial no.	Frame rate	1st Highest peak (x)	2nd Highest peak (y)	(x-y)/y
1	29.97	0.5	0.007	70.4
2	29.97	0.16	0.05	2.2
3	29.97	0.16	0.14	0.14
4	29.97	0.51	0.04	11.75
5	29.97	0.62	0.25	1.48
6	29.97	0.19	0.04	3.75
7	29.97	0.34	0.02	16



(a). Real Face video and its Pulse

Conclusion

- ❖ The field of CGI detection has seen many rapid strides in the last few years, and still research is being carried on.
- ❖ With this project we tried to create an arsenal of methods to deal with most of the computer generated forgeries like spams, CG images, rich texture images, face videos, and general videos.
- ❖ All the methods implemented had decent accuracy of 90% and above. The first method deals with SPAMs, second one deals with general CGIs and its extension deals with videos with CG scenes. The last paper detects forgery in faces present in videos.

References

[1] Hazza, Zubaidah Muataz, and Normaziah Abdul Aziz. "Detecting computer generated images for image spam filtering." Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on. IEEE, 2012.

[2]Tan, D. Q., et al. "Detecting computer generated images based on local ternary count." Pattern Recognition and Image Analysis 26.4 (2016): 720-725.

[3]Peng, Bo, et al. "Detection of computer generated faces in videos based on pulse signal." Signal and Information Processing (ChinaSIP), 2015 IEEE China Summit and International Conference on. IEEE, 2015.

[4]de Haan, Gerard, and Vincent Jeanne. "Robust pulse rate from chrominance-based rPPG." IEEE Transactions on Biomedical Engineering 60.10 (2013): 2878-2886.

[5] G. Schaefer and M. Stich, "UCID - An uncompressed colour image database," in Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia, San Jose, 2004.

[6] Image Spam Hunter, Yan Gao, Ming Yang and Xiaonan Zhao EECS Department, Northwestern University, http://www.cs.northwestern.edu/~yga751/ML/ISH.htm.