

# An Era of Face Filters

Swapnil Bhardwaj, Madhurima Hooda, Saru Dhir

**Abstract:** Smartphones are widely growing among today's generation and the main reason for it is the social media which helps connect people whenever and wherever they are in any part of this world. Social media apps like Snapchat got it's popularity because of it's unique features like face filters. Face filters are widely used among teenagers for fun and to look different. It has the ability to entertain by applying different types of filters on a person's face by recognizing it's facial features and make it look fun built caught popularity very fast and is growing every day. The augmented reality technology works flawlessly behind the filters. The technology first originated in Russia technical company which afterwards was overtaken by "snapchat" and grew constantly. The aim of this article is to create an application that can recognise a human face and its features and accordingly apply face filters. Using this feature user can view the content with great experience.

**Keywords:** snapchat, combining, matrix, powershell

## I. INTRODUCTION

This article approaches human face recognition and applying different filters to the human face. The augmented reality filters play an important role in the profligaterising arena of "computer vision". "Computer vision" is direct opposite of computer graphics. Usage of computer vision is growing day by day in our society and is becoming important in people's lifestyle. It is making the world more advanced it helps facebook to check who are the people in your group photo, it can also help self-driving car to avoid running over people. It is advancement is the human evolution. When humans see a face or a lot of faces at the same time, the brain processes a lot of information that will trigger the memory nerves and you will try to recall whether you know that person or not and will react accordingly. When a computer machine sees a face or any digital image, it will see a blank screen with different color codes on each pixel. Something like the photo shown below.



Fig. 1. Matrix of numerical digits

- This large matrix of numerical digits are codes, and each combination of these numerical digits represent a different colour and pixel.
- The face detection algorithm will go through this code and look for a colour patterns that would eventually represent a face.
- There are Different parts of the face that give different details. For example, the bridge of the nose is lighter than its surroundings. The under-eye skin is darker than the forehead, and the centre of the forehead is lighter than its other sides.
- Statistical model of a face can be created by manually pointing out different borders of the facial features with appointing numbers to the different regions of the face. When you take a phot of your face on the screen, these already pre-defined points will align themselves and will look for areas of contrast to know exactly where your, nose, lips, jaw line, eyes and eyebrows are located. This statistical model looks something like this.

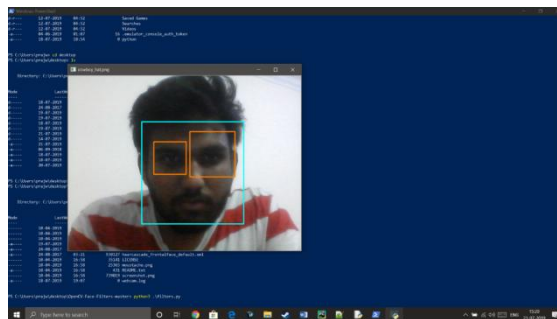


Fig. 2. Face detection algorithm working to detect different colours

## 1.2 CAMERA

A camera is a tool that captures different moments of our life and surrounding. We use camera for security, taking pictures, shooting videos,



Revised Manuscript Received on November 22, 2019.

\* Correspondence Author

Ms Swapnil Bhardwaj, B.Tech in Information Technology from Amity School of Engineering & Technology, Amity University, Noida,

Saru Dhir, Software Engineering, Agile Development, Software Testing, Cyber crime.

Madhurima Hooda, Assistant Professor in the Discipline of Information Technology at Amity School of Engineering and Technology, Amity University, Uttar Pradesh, Noida, India.

authentication and selfies. Today we use camera for the authentication, face detection and with other third-party software for security purpose. There are endless opportunities and features that we can create and make using the camera feature. It is the most secure authentication features in the devices after fingerprint sensor. Apple is widely using this feature to unlock their phones. Modern laptops use this feature to unlock the operating system. Like in windows 10 there is Windows hello feature.

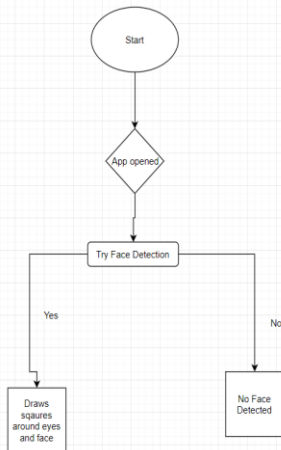


Fig. 3. Flow chart for face detection algorithm

## II. THE PROPOSED SYSTEM

### Face Detection –

Working of face detection is below:

In the given image of a group of people the computer recognises the faces of people and display an square box around the faces and two small rectangle boxes around the eyes with the X and Y coordinates.



Fig. 4. Detecting features by making rectangular boxes

“Face detection” was a very big problem in the early 2000s but researchers made it possible to solve it. There still persisted some more problems like the details of the facial features which was difficult for computer to understand and locate and was solved with the increasing improvement in machine learning [6] which detected good quality, but this method cannot be used in real time detection on CPU.

### Facial Landmarks –

Working of facial landmarks is as follows:

For each spotted face, it yields the resident area coordinates for individual member or facial feature of that specific face. This drives to identify the coordinates of the eyes, nose, lips, mouth and mouth. The coordinates are typically the combination of points (X,Y).

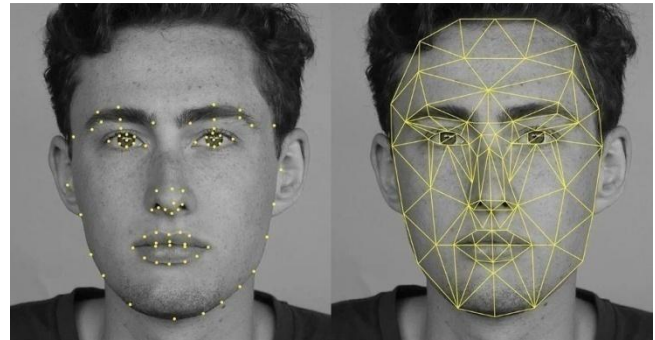


Fig. 5. Facial landmarks

Extracting facial landmarks is an easy process for the CPU in a bounding box (i.e. Cropped image with the target face), but it is quite problematic to device for the programmer. It requires machine learning techniques and the classifier is used for training and running.

### Image Processing -

Features applied used image processing features [7,8] on the full facial structure of human being. It is achieved with the help of Active Shape Model.

“Active Shape Model” is the model of facial features that uses thousands of photos to mark the borders of face and identify facial features.

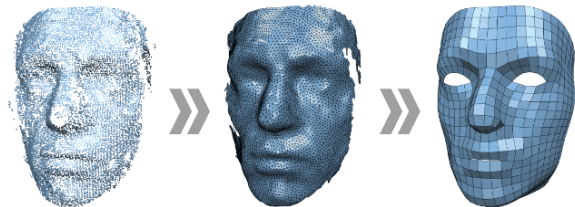


Fig. 6. Active Shape Model

With the help of machine learning, a face is created which then align with the face in the photo that was given to the computer to scan. People have diverse faces and is not possible that every face will fit according to the algorithm. Pixels are automatically fixed according to the face and cover all the edges. It is then have to look for difference in the shaded parts of the faces in the image or photo. Procedure that go through this training in the “machine learning” technique will have the basic features of the average “facial features”.

It is possible that the initial changes are wrong, and the algorithm works its magic and version the location of all extra points that it has fixed and will correct the boo-boos made. 3D model can be created which will scale the face and make a shift.

The whole process of ‘facial recognition’ is completed when the filter is chosen. The filters are not accurate and few parts of the face in the camera and adding some image on top of them which is selected by the user as the filter which is to be used.

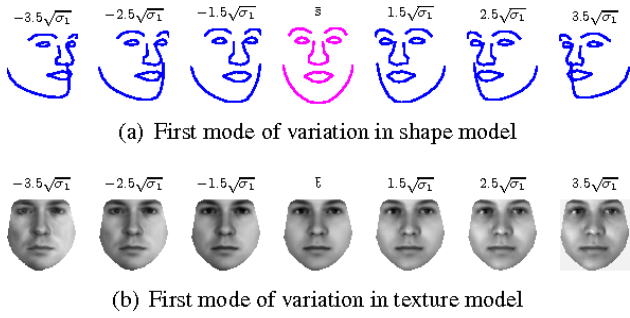


Fig. 7. Different variations

### III. WORKING

Below is the code written in python and was tested on python3 in PowerShell on windows 10. It used libraries cv2. The other xml files were used to detect the different parts of the face like eyes and the whole face structure. It will eventually create boxes around those features of the face to highlight it.

```

import cv2
import sys
import logging as log
import argparse as ap
from time import sleep
import numpy as np
import os
import subprocess

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('--input', type=str, help='Input image path')
    parser.add_argument('--output', type=str, help='Output image path')
    parser.add_argument('--model', type=str, help='Model path')
    parser.add_argument('--threshold', type=float, help='Threshold')
    parser.add_argument('--display', type=bool, help='Display the result')
    args = parser.parse_args()

    input_image = args.input
    output_image = args.output
    model_path = args.model
    threshold = args.threshold
    display = args.display

    # Load the model
    model = cv2.CascadeClassifier(model_path)

    # Load the image
    image = cv2.imread(input_image)
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Detect faces
    faces = model.detectMultiScale(gray, threshold)

    # Draw rectangles around the faces
    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y), (x+w, y+h), (255, 0, 0), 2)

    # Save the result
    cv2.imwrite(output_image, image)

    # Display the result
    if display:
        cv2.imshow('Face Detection', image)
        cv2.waitKey(0)
        cv2.destroyAllWindows()

if __name__ == '__main__':
    main()
    
```

Fig. 8. Face detection algorithm (python code)

```

import cv2
import sys
import logging as log
import argparse as ap
from time import sleep
import numpy as np
import os
import subprocess

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('--input', type=str, help='Input image path')
    parser.add_argument('--output', type=str, help='Output image path')
    parser.add_argument('--model', type=str, help='Model path')
    parser.add_argument('--threshold', type=float, help='Threshold')
    parser.add_argument('--display', type=bool, help='Display the result')
    args = parser.parse_args()

    input_image = args.input
    output_image = args.output
    model_path = args.model
    threshold = args.threshold
    display = args.display

    # Load the model
    model = cv2.CascadeClassifier(model_path)

    # Load the image
    image = cv2.imread(input_image)
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Detect faces
    faces = model.detectMultiScale(gray, threshold)

    # Draw rectangles around the faces
    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y), (x+w, y+h), (255, 0, 0), 2)

    # Save the result
    cv2.imwrite(output_image, image)

    # Display the result
    if display:
        cv2.imshow('Face Detection', image)
        cv2.waitKey(0)
        cv2.destroyAllWindows()

if __name__ == '__main__':
    main()
    
```

Fig. 9. Face detection algorithm (python code)

```

import cv2
import sys
import logging as log
import argparse as ap
from time import sleep
import numpy as np
import os
import subprocess

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('--input', type=str, help='Input image path')
    parser.add_argument('--output', type=str, help='Output image path')
    parser.add_argument('--model', type=str, help='Model path')
    parser.add_argument('--threshold', type=float, help='Threshold')
    parser.add_argument('--display', type=bool, help='Display the result')
    args = parser.parse_args()

    input_image = args.input
    output_image = args.output
    model_path = args.model
    threshold = args.threshold
    display = args.display

    # Load the model
    model = cv2.CascadeClassifier(model_path)

    # Load the image
    image = cv2.imread(input_image)
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Detect faces
    faces = model.detectMultiScale(gray, threshold)

    # Draw rectangles around the faces
    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y), (x+w, y+h), (255, 0, 0), 2)

    # Save the result
    cv2.imwrite(output_image, image)

    # Display the result
    if display:
        cv2.imshow('Face Detection', image)
        cv2.waitKey(0)
        cv2.destroyAllWindows()

if __name__ == '__main__':
    main()
    
```

Fig. 10. Face detection algorithm (python code)

Below is the hat and moustache filter in the working.

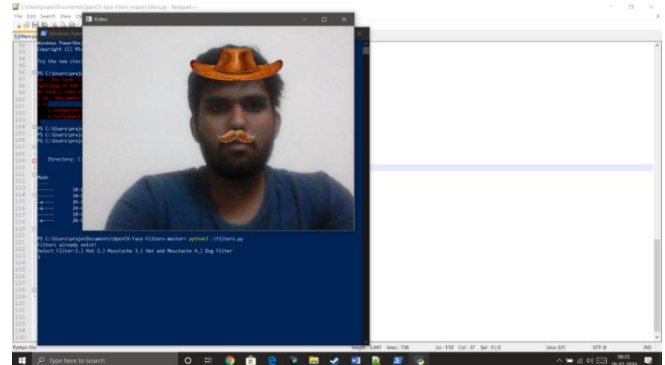


Fig. 11. Hat and moustache filter on image

Below is the camera log table and shows the entries.

Time	Process	Path	Size	Format	Resolution	Frame Rate
2019-07-26 11:28:04.709823	INFO:root:Faces: 1	at	2019-07-26 11:28:04.709823			
2019-07-26 11:28:04.808880	INFO:root:Faces: 2	at	2019-07-26 11:28:04.808880			
2019-07-26 11:28:04.918480	INFO:root:Faces: 1	at	2019-07-26 11:28:04.918480			
2019-07-26 11:28:05.028080	INFO:root:Faces: 2	at	2019-07-26 11:28:05.028080			
2019-07-26 11:28:05.137680	INFO:root:Faces: 1	at	2019-07-26 11:28:05.137680			
2019-07-26 11:28:05.247280	INFO:root:Faces: 2	at	2019-07-26 11:28:05.247280			
2019-07-26 11:28:05.356880	INFO:root:Faces: 1	at	2019-07-26 11:28:05.356880			
2019-07-26 11:28:05.466480	INFO:root:Faces: 2	at	2019-07-26 11:28:05.466480			
2019-07-26 11:28:05.576080	INFO:root:Faces: 1	at	2019-07-26 11:28:05.576080			
2019-07-26 11:28:05.685680	INFO:root:Faces: 2	at	2019-07-26 11:28:05.685680			
2019-07-26 11:28:05.795280	INFO:root:Faces: 1	at	2019-07-26 11:28:05.795280			
2019-07-26 11:28:05.904880	INFO:root:Faces: 2	at	2019-07-26 11:28:05.904880			
2019-07-26 11:28:06.014480	INFO:root:Faces: 1	at	2019-07-26 11:28:06.014480			
2019-07-26 11:28:06.124080	INFO:root:Faces: 2	at	2019-07-26 11:28:06.124080			
2019-07-26 11:28:06.233680	INFO:root:Faces: 1	at	2019-07-26 11:28:06.233680			
2019-07-26 11:28:06.343280	INFO:root:Faces: 2	at	2019-07-26 11:28:06.343280			
2019-07-26 11:28:06.452880	INFO:root:Faces: 1	at	2019-07-26 11:28:06.452880			
2019-07-26 11:28:06.562480	INFO:root:Faces: 2	at	2019-07-26 11:28:06.562480			
2019-07-26 11:28:06.672080	INFO:root:Faces: 1	at	2019-07-26 11:28:06.672080			
2019-07-26 11:28:06.781680	INFO:root:Faces: 2	at	2019-07-26 11:28:06.781680			
2019-07-26 11:28:06.891280	INFO:root:Faces: 1	at	2019-07-26 11:28:06.891280			
2019-07-26 11:28:07.000880	INFO:root:Faces: 2	at	2019-07-26 11:28:07.000880			
2019-07-26 11:28:07.110480	INFO:root:Faces: 1	at	2019-07-26 11:28:07.110480			
2019-07-26 11:28:07.220080	INFO:root:Faces: 2	at	2019-07-26 11:28:07.220080			
2019-07-26 11:28:07.329680	INFO:root:Faces: 1	at	2019-07-26 11:28:07.329680			
2019-07-26 11:28:07.439280	INFO:root:Faces: 2	at	2019-07-26 11:28:07.439280			
2019-07-26 11:28:07.548880	INFO:root:Faces: 1	at	2019-07-26 11:28:07.548880			
2019-07-26 11:28:07.658480	INFO:root:Faces: 2	at	2019-07-26 11:28:07.658480			
2019-07-26 11:28:07.768080	INFO:root:Faces: 1	at	2019-07-26 11:28:07.768080			
2019-07-26 11:28:07.877680	INFO:root:Faces: 2	at	2019-07-26 11:28:07.877680			
2019-07-26 11:28:07.987280	INFO:root:Faces: 1	at	2019-07-26 11:28:07.987280			

Fig. 12. Camera log table

### C. Steps to improve Face Filters

“Machine learning” is getting popular in today’s world as it is very useful in our daily lives. It enables the analysis huge part of the data. “Combining Machine learning and Artificial Intelligence and Cognitive technologies” will make it more practical in processing huge amount of information and data. It will help in more accurate reading of facial structure and will help improve the quality of face filters. Primary aim is to allow the computers to learn and adapt. If the computer perceives the face in the given image accurately, it will be able to use the filters more accurately and thus





showing the image more beautiful and sharper.

### IV. TESTING AND EVALUATION

The proposed system application was tested and evaluated on ASUS laptop on PowerShell with front camera, running windows 10. From this experiment is found that the performance of proposed application depends upon the processing power and the resolution of the camera and in addition to the intensity of the used light in the surrounding such as indoor or outdoor usage.

### V. SUMMARY

1. The augmented reality filters play an important role in the large and constantly growing domain of personal computer world.
2. For each detected face, it outputs the local region coordinates for each member or facial feature of that face. This will include the eyes, bone, lips, nose, mouth, coordinates usually are in the form of points.
  - a) Features applied used image processing features on the full facial structure of human being. It is achieved with the help of Active Shape Model.
  - b) "Active Shape Model" is the model of facial features that uses thousands of photos to mark the borders of face and identify facial features.
  - c) 3D model can be created which will scale the face and make a shift.
  - d) Algorithm that go through this training in the "machine learning" technique will have the basic features of the average "facial features".

### VI. CONCLUSION

This paper presented a proposed face filter application working on the open cv with python. The proposed application provides the ability of changing filters on face and looking different. The introduced application uses the built-in front camera of the laptop without the need of additional sensor and instruments. The obtained result is fast and efficient and makes the selfie camera more fun and efficient for people.

### REFERENCES

[1] Chung, K. C., Kee, S. C., & Kim, S. R. (1999, September). Face recognition using principal component analysis of Gabor filter responses. In *Proceedings International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems. In Conjunction with ICCV'99 (Cat. No. PR00378)* (pp. 53-57). IEEE.

[2] Savvides, M., Kumar, B. V., & Khosla, P. K. (2004, August). Cancelable biometric filters for face recognition. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* (Vol. 3, pp. 922-925). IEEE.

[3] Bhuiyan, A. A., & Liu, C. H. (2007). On face recognition using gabor filters. *World academy of science, engineering and technology*, 28, 51-56.

[4] Wang, Y., & Chua, C. S. (2005). Face recognition from 2D and 3D images using 3D Gabor filters. *Image and vision computing*, 23(11), 1018-1028.

[5] Štruc, V., & Pavešić, N. (2009, September). Principal gabor filters for face recognition. In *2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems* (pp. 1-6). IEEE.

[6] Chotwani, Priyal, Asmita Tiwari, and Madhurima Hooda. "Fraudulent Loan Prediction using Machine Learning Algorithms." *Indian Journal of Public Health Research & Development* 10.5 (2019): 845-850.

[7] Madhurima, Madhulika. "Object tracking in a video sequence using Mean-Shift Based Approach: An Implementation using MATLAB7." *IJCEM International Journal of Computational Engineering & Management* 11 (2011).

[8] Madhurima, Ravindra Purwar and Madhulika "A GUI for image segmentation using MATLAB" published in *International Journal of Intelligent Information Processing* ", Jan-June 2010, issue of the journal.

### AUTHORS PROFILE



Ms Swapnil Bhardwaj is pursuing B.Tech in Information Technology from Amity School of Engineering & Technology, Amity University, Noida, Batch (2016-20). Her area of interests are Machine Learning, Artificial Intelligence, Deep Learning, face recognition and Data Science. She has published and presented research papers in several conferences of repute. She has attended various conferences and seminars at national level.



Dr. Madhurima Hooda is currently working as an Assistant Professor in the Discipline of Information Technology at Amity School of Engineering and Technology, Amity University, Uttar Pradesh, Noida, India. She has approximately 14 years of Teaching and research experience. She has published many research papers in national and international journals and conferences of repute. She has published a book titled "Computer Networks" with Laxmi publications and contributed as an editor in global edition of the book "Absolute Java", Pearson Education. Her area of research is video object tracking, software testing, IoT, AJAX Applications and Machine learning. Her M.Tech work has been published as a book titled "Video Object Tracking" by LAP LAMBERT Academic Publishing GmbH & Co. KG, Germany. She has currently three patents and two copyrights on her name. She is an editor of *Journal of Information Processing Systems* (Scopus indexed). She is also a member of several professional bodies IETE, IFERP, IOAP and many more.



Dr. Saru Dhir has done her Ph.D in CSE, M.Tech in CS and M.Sc. She has more than 14 years of experience in teaching and research. Her research areas are: Software Engineering, Agile Development, Software Testing, Cyber crime. She has published many research papers in national and international journals and conferences of repute. She has completed Microsoft technology associate certification in DBA, C#, Networking and visual basic.