Ministry of Science and Higher Education of the Republic of Kazakhstan
L.N. Gumilyov Eurasian National University

Faculty of Information Technology
Department of Information Systems

MASKS, FILTERS AND EFFECTS ON THE FACE

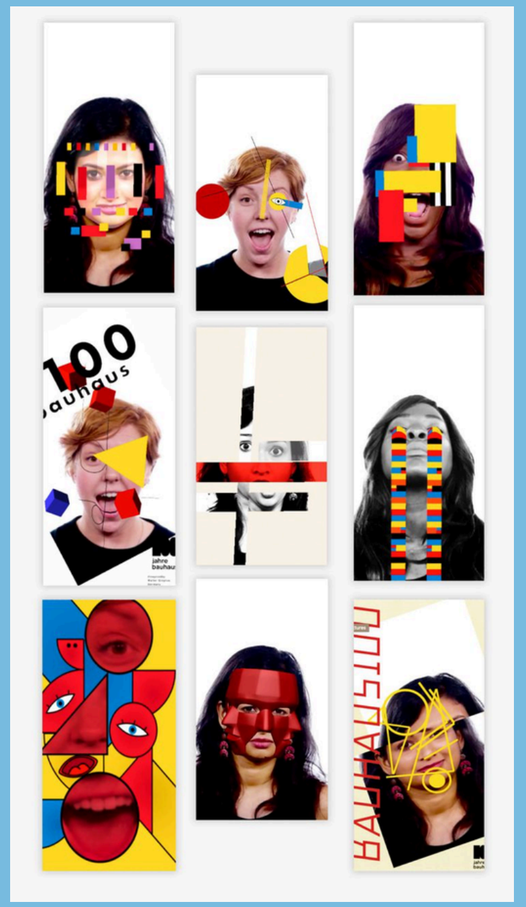
Done by: Zhumabek A.R
Check: Zhukabaeva T.K

DEFINING MASKS, FILTERS AND EFFECTS

Masks are virtual stencils that are applied to a layer and can be disabled or modified at any time, separate from the original contents of the layer. Masks are used to hide or display certain parts of an image.

Filters are tools that allow you to quickly change the appearance of an image. For example, you can apply a blur filter to soften an image or a sepia effect to give it a vintage look.

Effects are a variety of ways to diversify images or videos. For example, effects can distort an image, change the background, or draw something on it.



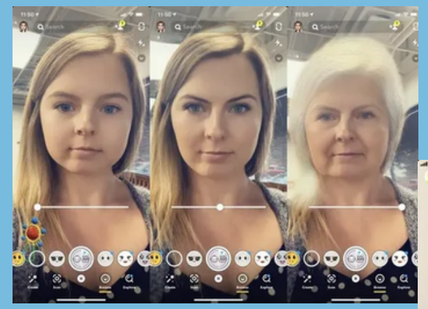
WHERE THEY ARE USED

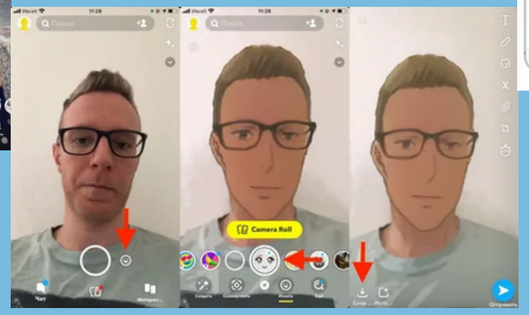
Masks, filters and facial effects are widely used in social media, games, video calls and augmented reality (AR) applications.

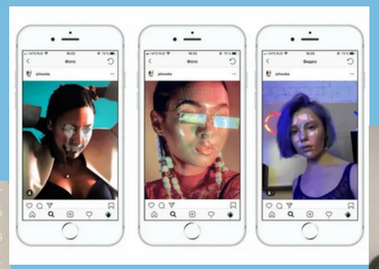
The main purpose of such technologies is to change appearance, add decorative elements, stylize an image or improve video quality.

The main areas of focus are:

- 2D filters: graphics overlay, color filters, blurring.
- 3D masks: animation effects, virtual cosmetics, avatars.
- Neural network stylizations: changing appearance using generative networks.
- Augmented Reality (AR): dynamic effects with real-time facial tracking.







FACE DETECTION AND TRACKING

Viola-Jones is a classical face detection method based on Haar cascades.
Advantages: high speed.
Disadvantages: sensitivity to illumination, low accuracy.

HOG + SVM - a method that uses gradient histograms to extract facial contours.
Works well for static images, but poorly suited for video.

Modern neural network models:

- MTCNN (Multi-task Cascaded Convolutional Networks) - works well with different perspectives.
- MediaPipe Face Detection - fast and accurate tool from Google for real-time face tracking.
- YOLO (You Only Look Once) - suitable for detecting multiple faces in an image

RECOGNIZING KEY FACIAL LANDMARKS

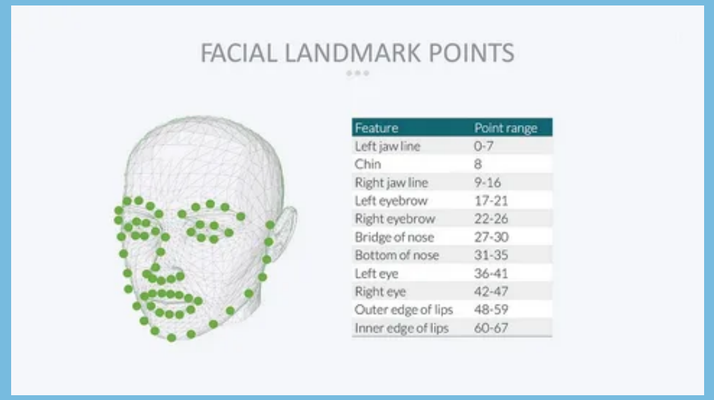
Facial Landmarks is a method for recognizing key points on the face, used to track facial expressions and position.

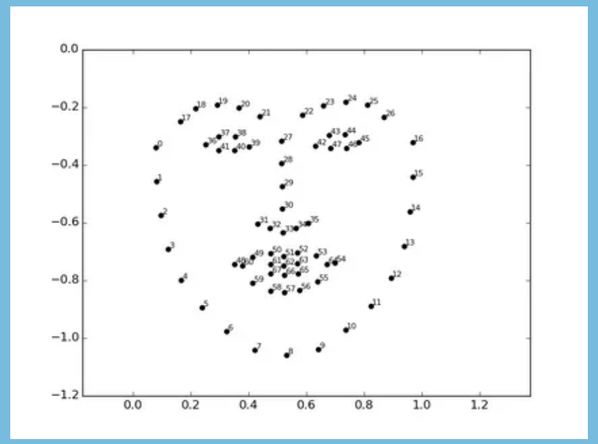
Dlib (68 points) - a popular facial markup tool:

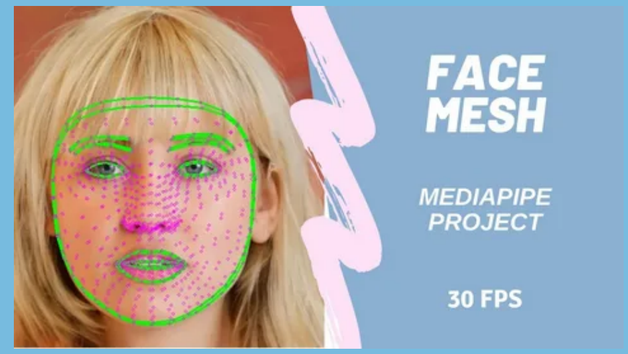
- Highlights eyebrows, eyes, nose, lips, and facial contour.
- Used for filters, animation, and emotion analysis.

MediaPipe Face Mesh (468 points) - Advanced model for 3D animation and masks:

- Allows you to build highly accurate 3D face models.
- Used in AR applications.







APPLICATION OF MASKS AND FILTERS

2D Filters:
Simple graphic effects (glasses, mustache, makeup) applied on top of an image.
Used in Snapchat, Instagram.

Face Warping:
Morphing and affine transformations are applied.
Comical effects (facial stretching, shrinking) are created.

3D Masks:
Binding 3D objects to the face using Blender, OpenGL, ARKit.
Creating dynamic masks (moving avatars, virtual hairstyles).

APPLICATION OF NEURAL NETWORKS

StyleGAN, DeepFake - generative models for appearance modification.

- Allow face replacement, image stylization.

CNN for image stylization:

- Used in applications like DeepArt and Prisma.
- Allows you to turn photos into drawings of different styles.

Face segmentation:

- Used to replace the background.
- Used in virtual makeup and beauty filters.

CONCLUSION

Facial processing technologies continue to evolve rapidly.
More realistic and interactive AR effects are possible in the future.
Application covers not only entertainment, but also professional spheres:

- Medicine: facial expression analysis, disease diagnosis.
- Cybersecurity: protection against DeepFake and facial spoofing.
- Virtual avatars: use in meta-universes and video games.

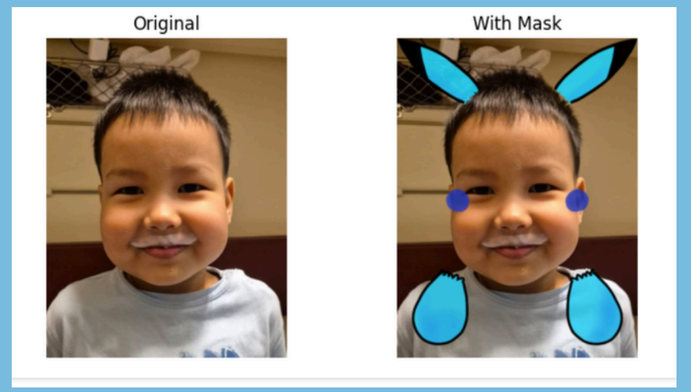
This code performs a transparent mask overlay (with effects) on a face image using OpenCV and Matplotlib.

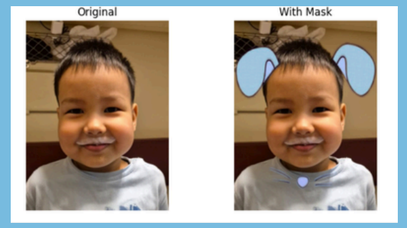
```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

image_path = "/content/11.jpg"
mask_path = "/content/eff1.png"
image = cv2.imread(image_path, cv2.IMREAD_COLOR)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
mask = cv2.imread(mask_path, cv2.IMREAD_UNCHANGED)
mask = cv2.resize(mask, (image.shape[1], image.shape[0]))
def apply_mask(image, mask):
    if mask.shape[2] == 4:
        alpha_mask = mask[:, :, 3] / 255.0
        mask_rgb = mask[:, :, :3]

        for c in range(3):
            image[:, :, c] = (1 - alpha_mask) * image[:, :, c] + alpha_mask * mask_rgb[:, :, c]
    return image
masked_image = apply_mask(image.copy(), mask)
plt.figure(figsize=(8, 4))
plt.subplot(1, 2, 1)
plt.imshow(image)
plt.title('Original')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(masked_image)
plt.title('With Mask')
plt.axis('off')
```





Result we see graph with two images:

Original face image.
Image with an overlaid mask/effect.

This code can be adapted for real time using OpenCV and the video stream from the camera.

THANK YOU