Face recognition and blurring

Zulkhairatuly Yerlan

1. Background

In today's world, privacy is an important issue, especially in video and photo materials. Face blurring is one of the methods of protecting personal data, widely used in video surveillance systems, journalism and social networks.

Key questions:
How to automate the process of face detection and blurring?
Which detection algorithms are the most accurate and fastest?

2. Objectives

Main goal: Develop a system capable of automatically detecting and blurring faces in real time.

Research and compare face detection algorithms (Haar Cascade, MTCNN, DeepFace).
Implement several blurring methods (Gaussian Blur, Pixelation, etc.).
Develop a user-friendly graphical interface (GUI) for working with the system.

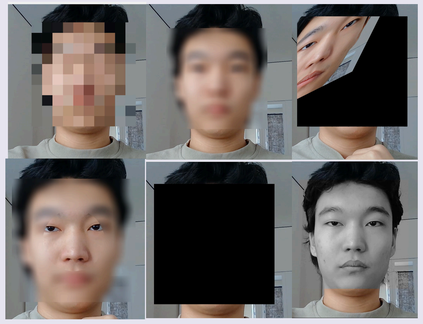
3. Methods

Face Detection
Haar Cascade: Classic algorithm using Haar cascades to detect faces.
MTCNN: Deep neural network that provides more accurate face detection.
DeepFace: Modern deep learning-based method that provides better performance.

Face Blur:
Gaussian Blur.
Pixelization.
Black Box.
Emoji detection.

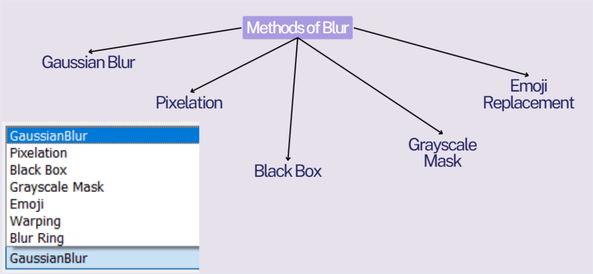
3a. Participants

Haar Cascade – Fast, but less accurate.
MTCNN – Good balance of accuracy and speed.
DeepFace – Most powerful, but slow.



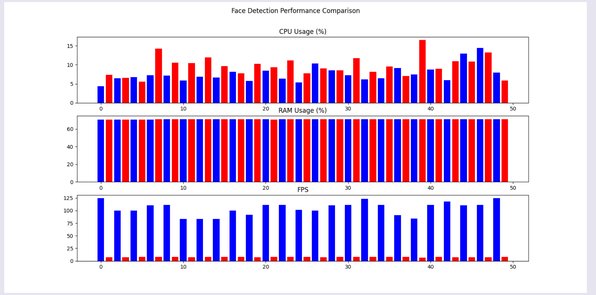
Parameter	Haar Cascade	MTCNN	DeepFace
Accuracy	Average	High	Very High
Speed	Higjh	Average	Low
Light resistance	Low	Average	High
Works with emotions?	No	Particula rly	Yes

3b. Analysis



4. Results

Automatic blurring of faces in real time is successfully implemented.
Different detection methods are tested and compared.
GUI allows users to easily select detection algorithms and blurring methods.
System performance depends on the chosen method: MTCNN gives the best balance of accuracy and speed.
DeepFace provides the best results, but requires more resources.



The graph compares Haar Cascade (blue) and MTCNN (red) based on three metrics. CPU Usage (%): MTCNN consumes significantly more CPU than Haar Cascade, which is expected due to its deep learning-based approach, whereas Haar Cascade relies on simpler feature-based detection. RAM Usage (%): Both methods show similar memory consumption, maintaining a range of approximately 60–70%, suggesting that RAM is not a major differentiating factor between them. FPS (Frames Per Second): Haar Cascade runs significantly faster, achieving much higher FPS compared to MTCNN, which operates at a much lower frame rate due to the computational complexity of deep learning-based face detection. Conclusion: Haar Cascade is more efficient in terms of speed and CPU usage, making it a better choice for real-time applications on resource-limited hardware, while MTCNN, despite being slower, may offer better accuracy and robustness in complex scenarios.

5. Conclusion

An efficient face blurring system with a choice of algorithms has been implemented.
Haar Cascade is suitable for fast analysis, MTCNN for more accurate analysis, DeepFace for complex tasks.
Prospects: improving processing speed, introducing new blurring methods, and improving performance on mobile devices.

References

1. Viola, P., & Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
2. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks (MTCNN). IEEE Signal Processing Letters.
3. OpenCV Library – https://opencv.org/
4. DeepFace Library – https://github.com/serengil/deepface
5. PyQt5 Documentation – https://www.riverbankcomputing.com/static/Docs/PyQt5/
6. Face Anonymization Methods: A Survey. IEEE Transactions on Information Forensics and Security, 2022.

Github link:

Contact details

Zulkhairatuly Yerlan
050524551510@enu.kz