UDC 004.42

A. Shchur, O. Polshakova

MODERN TECHNOLOGIES FOR HIDING PEOPLE'S FACES USING OBJECT TRACKING BASED ON YOLOV5 AND DEEPSORT

Abstract: The object of study is a system for automated blurring of human faces in video. This article provides a detailed overview of modern technologies and principles of tracking objects in video with assigning them unique elements. Since most video editors still leave most of the work to the user, it was decided to optimize this process.

The aim of this work is to reduce the time spent on the process of hiding human faces in video files. To achieve this goal, it is proposed to use a modern detector - the YOLO convolutional neural network and the DeepSORT object tracking algorithm, which uses classical approaches to filtering input data and predicting the position of an object in space, as well as a modern neural network capable of distinguishing between people's faces.

As a result of this work, among free analogues on the Internet, the acceleration of face blurring was achieved up to 20%, which is a pretty good result.

Keywords: neural network, object detection, object recognition, detector, YOLO, DeepSORT, Kalman filter

Description of the problem

The problem of anonymity on the Internet has become particularly acute for us today. In times of war, the enemy can use information from videos or photos to their advantage. You can find a lot of information about a person's face, as modern search engines can extract information from social networks quite accurately using photo search. Of course, there are also knowledgeable users who try to hide data, but there are still a large number of people who do not have Internet security skills. Today, quite a lot of different photo material, open interviews, and video data is being broadcast, especially with military personnel, which puts them, their fellow soldiers, and even relatives and friends at risk. In their civilian lives, media personalities run their own video blogs, simultaneously filming other people who are inadvertently caught in the frame. Many of them do not pay attention to this, but there are those who do not want their face to be on any video.

The best way to avoid the dissemination of personal information of this kind is to hide the person's face. Usually, modern editors such as DaVinci Resolve, Filmora, Adobe Premiere Pro offer the user to manually process the video, keeping track of each person, but if there are many faces, this approach is irrational, as it will be a very long procedure. Therefore, the question arises of automating this process by transferring this task to a computer. Nowadays, neural networks have developed quite well, and they can quickly and accurately find the location of the desired objects in a photo or video. The latest technologies can significantly speed up the performance of various tasks, such as the task of hiding a person's face.

In general, the process of "blurring" or blurring to hide objects from the user's view is based on the principle of selecting an area with the desired object and following it. Thus, the program inserts a blur effect on the selected area on each frame. Usually, in video, the objects to be hidden are human faces. The problem is that in most cases there can be quite a few such objects in the frame and manually selecting each face may not be efficient in terms of user time. A way to improve this situation is to develop a functionality for finding all faces and recognizing them. The user will only have to choose which faces should be hidden and which should be left unchanged.

There is also a more automated approach that uses neural networks to recognize faces in each frame and link them to previous frames. However, it should be noted that this approach requires comparing each face with each other in two frames, which makes this algorithm inefficient, as such an operation requires a lot of computing resources and time, and becomes significantly more complicated as the number of faces grows.

In this article, we propose to consider a modern approach to solving the problem of hiding a person's face by tracking objects in video using Object tracking and deep machine learning technologies, which will reduce the time required to hide a person's face by blurring it.

Review of literature and methods for solving the problem

Object tracking is an approach that solves the problem of not only finding a certain object in a video, but also tracking the change in the position of this object during the video.

The main component of tracking is an object detector, since in order to get information about the location of an object in the image, it must be found. Next, the SORT algorithm or Simple Online and Realtime Tracking1 is used to link one object between frames. For a better understanding of the entire algorithm that will allow face tracking, let's look at each of its components separately.

1. Object detector

Since video consists of individual frames (photos), the task of finding an object in a photo arises. Convolutional neural networks (CNNs) are best suited for this purpose. Such neural networks are built on the principle of a multilayer perceptron. The idea of developing such networks is biologically motivated and based on the connection scheme of the animal visual cortex [2].

This class of networks is perfect for image processing, as the network is resistant to various distortions, rotations, and movements of target objects when analyzing them. For example, a face may not always appear in the same place in a photo, it may be rotated or tilted. Also, a human face can express different emotions and change in every possible way between different frames.

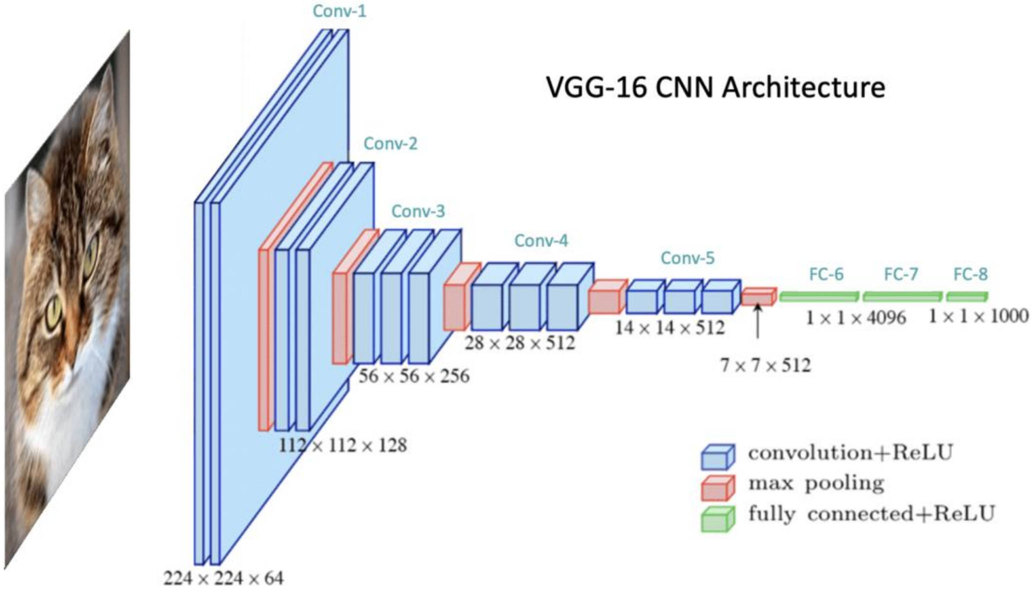

Fig. 1. Architecture of convolutional networks

To detect and recognize the necessary information in an image, popular but no less effective neural networks are used today. Examples of such networks that help to detect an object in a photo are YOLO and RCNN networks. Let us consider them in more detail.

RCNN (Regions with CNN) is an object detector that uses a combination of regions to potentially detect objects of interest in them and a convolutional neural network that processes the proposed regions and tries to find the specified objects in them [3].

The R-CNN algorithm follows:

– Generation of sample regions created using a selective search algorithm.

– Transfer of the generated sample regions that may contain the target object to the convolutional neural network, and any convolutional neural network architecture can be used here (VGG, AlexNET, etc.).

– Extracting features from each transmitted region using CNN.

– Transfer of the found features to a set of SVM (Support Vector Machine) classifiers.

– Localize a detected object using simple linear regression or bounding box regression.

RCNN training is essentially the training of a convolutional neural network on a huge data set that is pre-labeled. Also, while training this network, SVM classifier and linear regression are trained in parallel. One of the main disadvantages of the RCNN network is the time it takes to detect an object in a photo, since the algorithm involves processing each sample region.

In recent years, YOLO (You Only Look Once) networks have achieved great success in finding objects in photos or videos. In 2023, Ultralytics released the eighth version of the

YOLO network. This network has shown a significant speedup over previous versions, as well as improved accuracy.

YOLO is much faster than other similar neural networks because it runs only once for the entire input image, which is why it has its name [4].

The YOLO algorithm is as follows:

– Splitting the input image into chunks.

– Passing the cells with the parts of the image for which these cells are responsible on to the convolutional neural network.

– For each cell, a prediction is made regarding the classes to which it belongs.

– Final filtering of network response using thresholds.

YOLO also has its drawbacks. Since this network uses a fairly large grid, it may not be able to detect small objects very accurately. However, this is not a significant problem for the task at hand, because if the face in the photo is small in itself, it will be difficult to extract any information from it. As for comparing YOLO with other types of CNN models, such as RCNN, according to Priya Dwivedi's research [5], the 5th generation of YOLO significantly outperformed RCNN, it was both more accurate and faster. That is why we chose Ultralytics. Fig. 2-3 show a comparison of YOLO with other popular networks for finding objects in images.

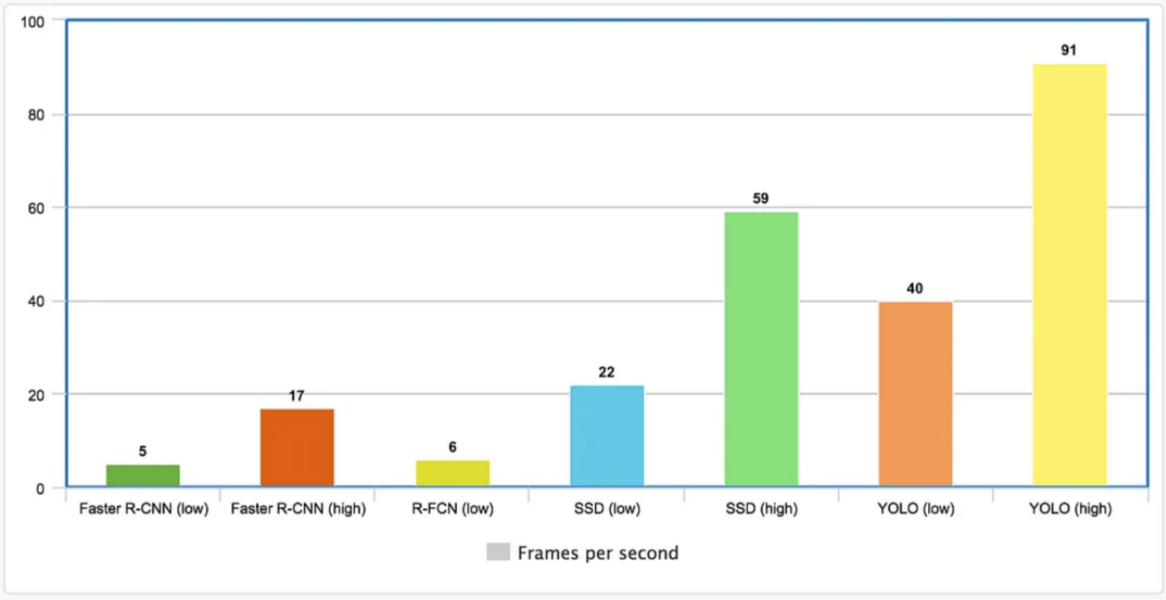

Fig. 2. Comparison of YOLO performance with popular detectors

From the figures above, we can see that YOLO networks have a good balance of accuracy and speed. Since you will have to process video, each second of which will contain 24 frames or more, speed is essential, but accuracy should be sufficient.

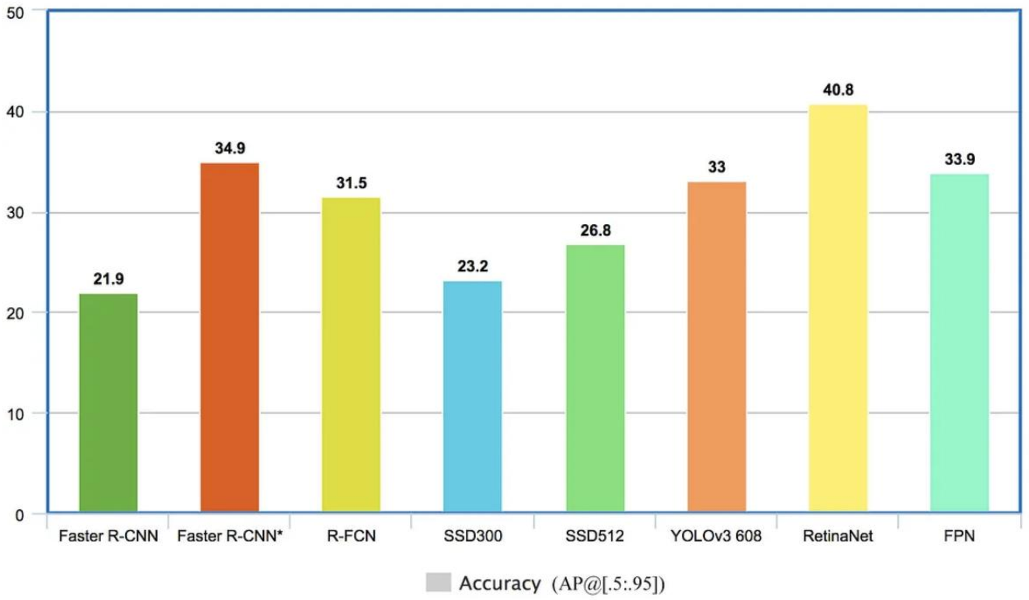

Fig. 3. Comparison of YOLO accuracy with popular detectors

2. Algorithm for tracking the positions of objects in the video

The development of tracking algorithms began with the development of SORT (Simple Online Realtime Tracking) [1]. This algorithm is based on the Kalman filter, which allows you to approximate a function and predict the next value based on information from previous records. The big advantage of the Kalman filter is that it improves its accuracy during the forecasting process. SORT uses data from previous frames about the position of target objects and predicts their further location. The tandem of detector and tracker allows you to assign uniqueness to objects in the video.

The next step was to improve this algorithm by adding an element of appearance. A network that could distinguish people by their appearance formed the basis of the DeepSORT algorithm [6].

To understand how the object tracker works, you need to understand what the Mahalanobis distance is and what the Kalman filter is.

2.1 Distance of Mahalanobis

Consider the problem of classifying two points using the distance from the point itself to some data distribution (Fig. 4).

If we were to use the usual Euclidean distance, point 1 and point 2 would have the same distance from the mean of the distribution, but the figure shows that point 1 is more closely related to the data distribution than point 2. Therefore, it is appropriate to use a metric that can evaluate not only the actual distance, but also the data distribution.

The Mahalanobis distance shows not only the distance in Euclidean space, but also takes into account the distribution of the data. In the task of tracking an object, it may happen that the object will change its position abruptly, or another object will enter the frame. Therefore, it is important to be able to distinguish objects not only by the actual distance from their previous position, but also to take into account the variance and covariance between them and their previous positions [7].

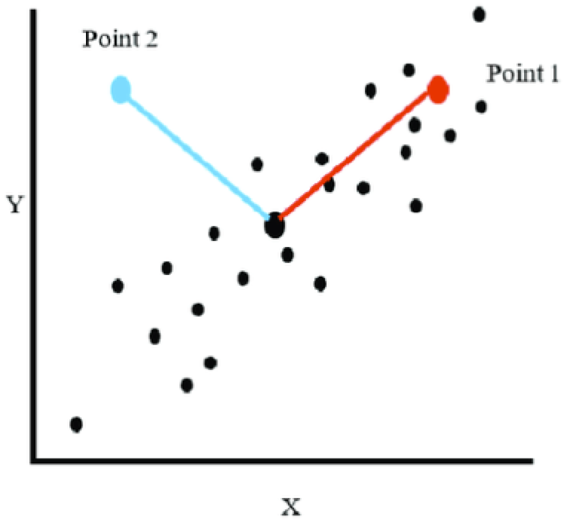

Fig. 4. Measuring the distance between objects in space

2.2 Kalman filter

This is a recursive filter that estimates the state vector of an object using information about its previous and current states. The advantage of this filter is that it improves its results over time, i.e. the more measurements are taken, the more accurate this filter works. It is very well suited to the task of determining the location of an object in a video by taking into account information about its previous position [8].

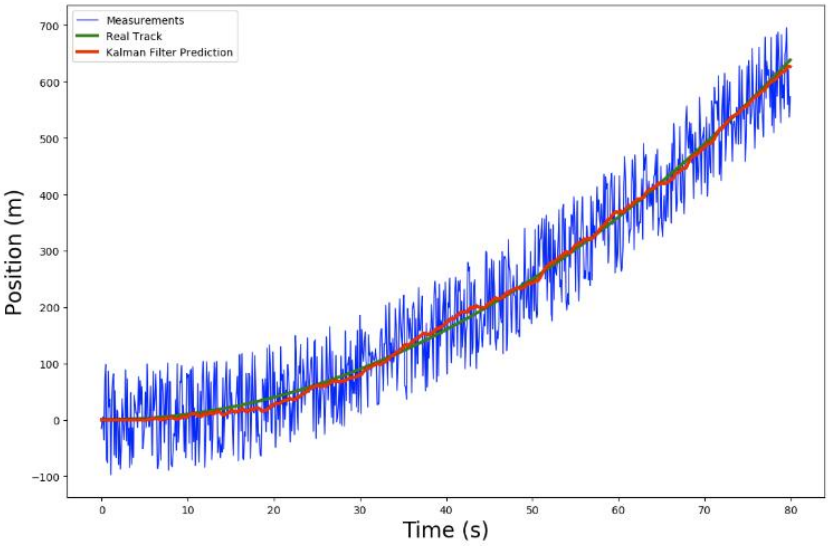

Fig. 5. Kalman filter operation with noisy data

2.3 SORT

SORT [1] is an algorithm that combines data from the detector about the object's location with the Kalman filter and the Mahalanobis distance. First, the detector receives a frame from the video and tries to find all the necessary objects. Then, the data is passed to the Kalman filter, which, based on the previous data on the positions of the objects, makes a prediction of where they will be in the current frame. Next, the Mahalanobis distance takes into account the distribution of the data and gives the class to which a particular object belongs. All this together makes it possible to distinguish objects from each other and assign them unique identifiers. In this way, you can get all the data about the location of an object for a certain number of video frames and let the system know that it is one object with the same identifier. Figure 6 shows the structure of the SORT algorithm.

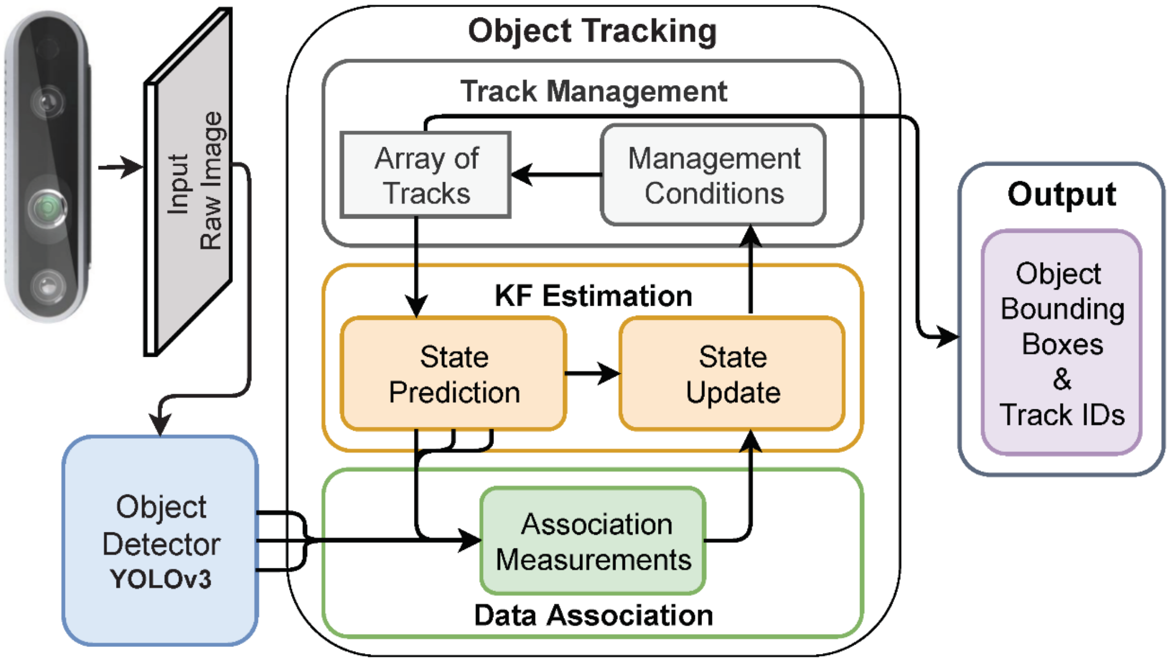

Fig. 6. The SORT algorithm

2.4 DeepSORT

When using a conventional SORT algorithm, a problem arises if two objects begin to cover each other in the video, then with this approach it will be impossible to distinguish between these objects, then it is possible that an identifier is assigned to the wrong object, which results in inaccuracy in the work. DeepSORT introduces an additional concept of "Appearance", i.e. it analyzes how the object looks like. This was done with the help of a neural network that was trained to distinguish people by their appearance [6].

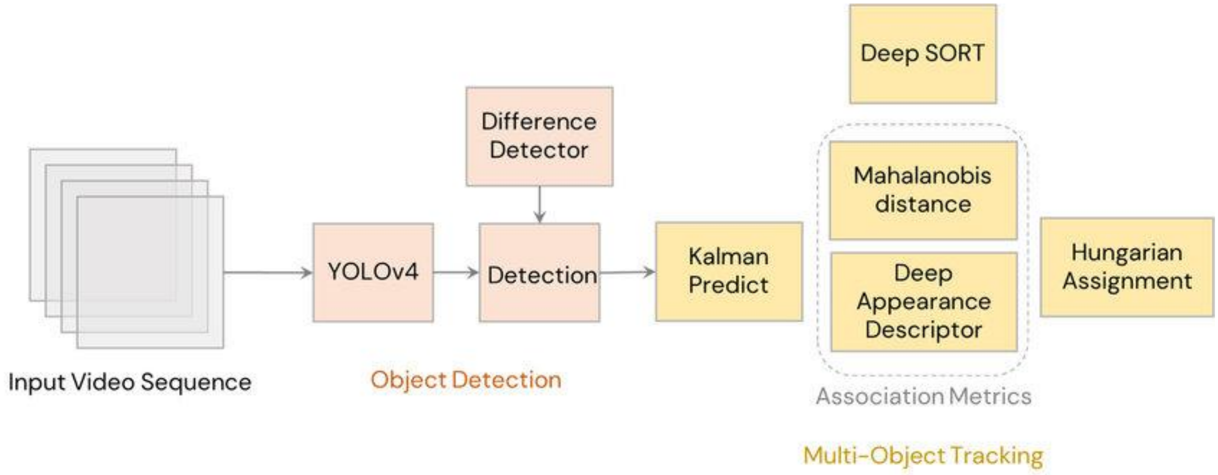

Fig. 7. DeepSORT algorithm

Practical results achieved

Since the purpose of this work was to find ways to speed up the process of hiding people's faces in video, it is advisable to compare the proposed algorithm with analogs available on the Internet. It should be noted that this algorithm can work both with the use of the CPU and with hardware acceleration based on video cards supporting Nvidia CUDA technologies. Therefore, we conducted experiments using both a CPU and a GPU.

4.1 Experiment №1. Blurring one of the two faces in the video

For this experiment, a video was downloaded. The duration of the video file is 4866 seconds. The results are shown in Tabl.1

Table 1. **Experiment 1. Blurring one of the two faces**

Input video file duration, seconds	«FaceAutoBlur»		Analog №1. «Facit Identity Cloack»	Analog №2. «Gallio»
	Computing device			
	CPU time, seconds	GPU time, seconds	CPU time, seconds	CPU time, seconds
2649	4866	2923	5198	5937

As we can see from the table above, the developed software processes video faster than the proposed analogues. If we consider the use of a graphics accelerator, the performance increases even more. The acceleration for processing on the CPU reaches 20%. When using a video card, the proposed solution works twice as fast.

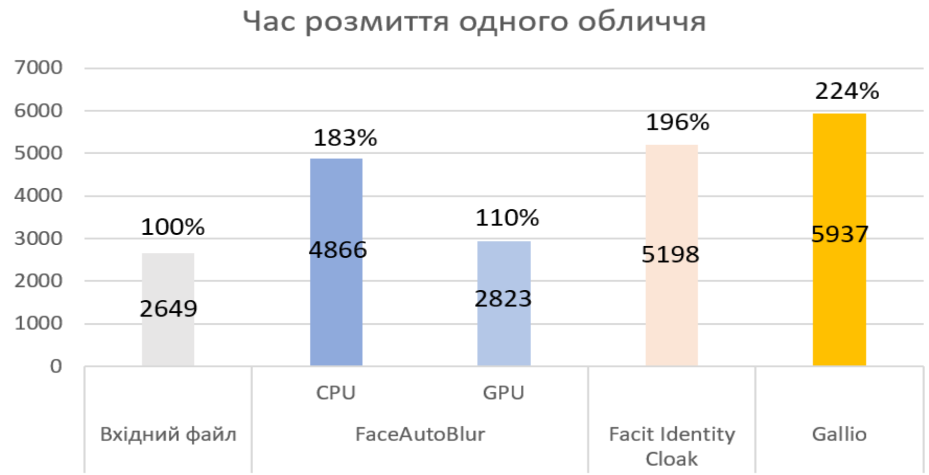

Fig. 8. Results of experiment №1

4.2 Experiment №2. Blurring two faces in a video

For this experiment, a video file from the previous experiment was used.

Table 2. **Experiment 2. Blurring all faces in the video**

Input video file duration, seconds	**«FaceAutoBlur»**		**Analog №1. «Facit Identity Cloack»**	**Analog №2. «Gallio»**
	Computing device			
	CPU time, seconds	**GPU time, seconds**	**CPU time, seconds**	**CPU time, seconds**
2649	4653	2754	4876	5724

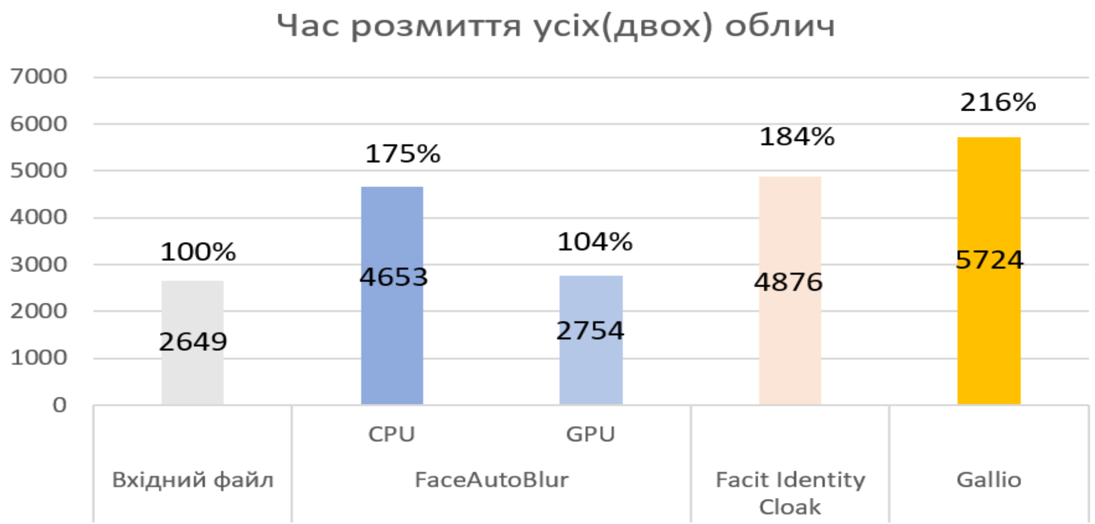

Fig. 9. Results of experiment №2

4.3 Experiment 3. Blurring four faces in a video

To conduct this experiment, the video was downloaded and its duration was reduced to 44:09 minutes, for greater clarity with the results of previous experiments.

Table 3. **Experiment 3. Blurring four faces in a video**

Input video file duration, seconds	**«FaceAutoBlur»**		**Analog №1. «Facit Identity Cloack»**	**Analog №2. «Gallio»**
	Computing device			
	CPU time, seconds	**GPU time, seconds**	**CPU time, seconds**	**CPU time, seconds**
2649	5408	3245	5602	6976

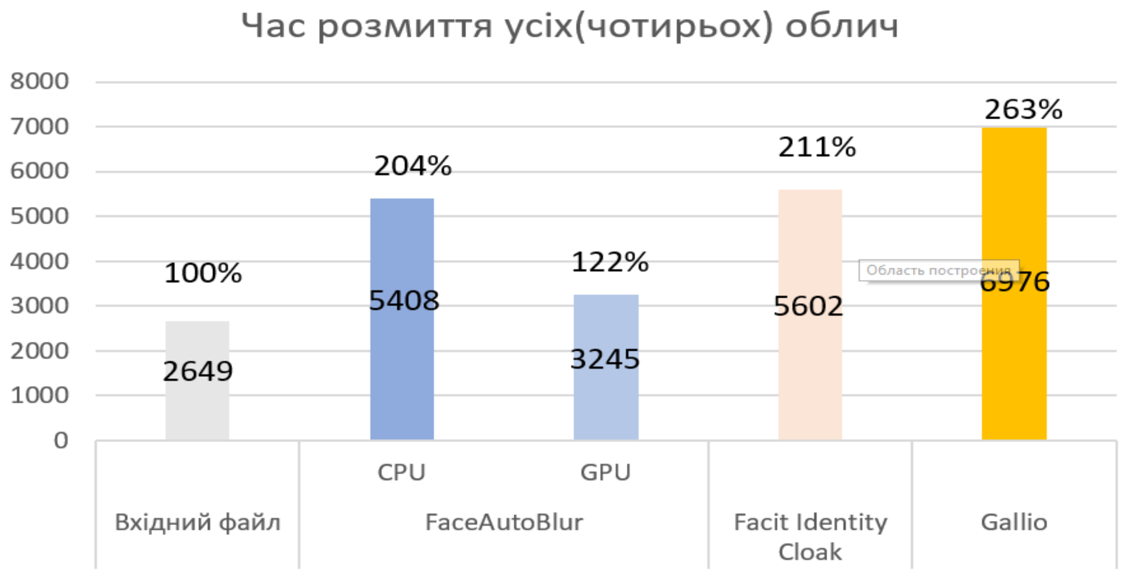

Fig. 10. Results of experiment №3

The results of the experiment show that the blurring time for more faces increased slightly.

Conclusions

As a result of the research, it was possible to transfer a significant part of the work of hiding human faces from humans to computers. Using modern deep machine learning technologies, it was possible to speed up this process compared to analogs available on the web.

The YOLO-based detector proved to be a great success, and with sufficient accuracy, it quickly detects the specified objects in the video. Its work in conjunction with the advanced DeepSORT tracker made it possible to accurately and quickly find faces in the video and assign them unique identifiers, thereby facilitating the user's work. The advantage of this algorithm is that it can work using the hardware capabilities of Nvidia graphics cards that support CUDA technology. Using the graphics accelerator, it was possible to process video in real time, which is much faster than similar solutions.

The implementation of this solution will definitely simplify the work of video editors and save their precious time.

REFERENCES

1. *A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft*, "Simple online and realtime tracking," 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 2016, pp. 3464-3468, doi: 10.1109/ICIP.2016.7533003.
2. *Zewen Li, Wenjie Yang, Shouheng Peng, & Fan Liu*. (2020). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. URL: https://arxiv.org/ftp/arxiv/papers/2004/2004.02806.pdf
3. *Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, & Junwei Han*. (2021). Oriented R-CNN for Object Detection. URL: https://www.arxiv-vanity.com/papers/2108.05699/
4. *Joseph Redmon, Santosh Divvala, Ross Girshick, & Ali Farhadi.* (2016). You Only Look Once: Unified, Real-Time Object Detection. URL: https://arxiv.org/pdf/1506.02640.pdf
5. *Priya D.* YOLOv5 compared to Faster RCNN. Who wins? URL: https://towardsdatascience.com/yolov5-compared-to-faster-rcnn-who-wins-a771cd6c9fb4
6. *Nicolai Wojke, Alex Bewley, & Dietrich Paulus*. (2017). Simple Online and Realtime Tracking with a Deep Association Metric. URL: https://arxiv.org/pdf/1703.07402.pdf
7. *McLachlan, G.J.* Mahalanobis distance. *Reson* **4**, 20–26 (1999). https://doi.org/10.1007/BF02834632
8. *Yan Pei, Swarnendu Biswas, Donald S. Fussell, Keshav Pingali.* (2019). An Elementary Introduction to Kalman filtering URL: https://arxiv.org/pdf/1710.04055.pdf