

## BACKGROUND MODELING AND FOREGROUND OBJECT DETECTION WITH ARTIFICIAL INTELLIGENCE

---

**Saida BEKNAZAROVA,**

*Doctor of technical Sciences, Professor, Tashkent University of Information  
Technologies named by Muhammad Al-Khwarizmi, Tashkent, Uzbekistan*

*E-mail: [saida.beknazarova@gmail.com](mailto:saida.beknazarova@gmail.com)*

**Abduvali MUKHAMMADIYEV,**

*Doctor of physical-mathematical Sciences, Professor, Tashkent University of  
Information Technologies named by Muhammad Al- Khwarizmi,  
Tashkent, Uzbekistan*

**Doston XAMROYEV,**

*student of Tashkent University of Information Technologies named  
by Muhammad Al- Khwarizmi, Tashkent, Uzbekistan*

**Islom ESHTEMIROV,**

*magistr of Tashkent University of Information Technologies named  
by Muhammad Al-Khwarizmi, Tashkent, Uzbekistan*

*DOI: <https://doi.org/10.47689/978-9943-7818-0-1-pp70-73>*

Image Enhancement is the process of improving the image quality without loss of information until the desired visual result (resolution, color and style) is obtained or the photo is prepared for further analysis in various computer vision programs: object recognition, classification, image interpretation. Image quality improvement usually includes a number of transformations: noise reduction, improvement of blurred photos, increase in resolution, contrast, lightening of dark photos, elimination of optical distortions, etc [2].

Image post-processing is an integral and important part of the photo creation process. It is needed to eliminate common disadvantages of photographs, including:

- underexposure;
- overexposure;
- low contrast;
- contour lighting of the shooting object;
- incorrect color balance;
- defocus and much more.

Popular photo editors, such as Adobe Photoshop, Adobe Lightroom or Raw Therapee, offer various sets of tools for retouching images and have been leading the market for a long time. Nevertheless, the quality of the processing result strongly depends on the skills and aesthetic perception of retouchers, and it is subjective. In addition, editing digital photos manually usually takes a lot of time [1].

Advantages of AI-based automatic photo processing

Modern photo auto-correction services greatly facilitate and simplify the retouching process for all users. Photo editors based on artificial intelligence do all the same things that a retoucher would do manually in Photoshop, and at the same time allow you to fully control the process. Automatic retouching helps:

- To give pictures taken with a simple camera a professional look and improve image quality;

- save time by allowing the program to do all the work instead of manual editing;
- to print and publishing companies - to improve the process of color correction of images for magazines, marketing campaigns and not only.

Image quality improvement algorithms are not limited to digital photography. They are used for processing medical images, for example, in radiology, where AI not only helps to improve the quality of the image itself, but can also detect certain diseases. We have already written about AI in medicine here.

“Smart” image quality improvement gives good results, which can be used in various computer vision systems in the future:

- improved infrared imagery is extremely important for applications, Smart City (Smart city);

- programs for handling video and photo forensics, for example, Amped FIVE allows to restore the shape of the car in motion, to measure the objects on the photographs from the crime scene, detect the faces blurred/dark photos, to improve the clarity of text on the photos we have seen this in the films about the FBI and CSI;

- when you need to sharpen a photo in conditions of haze and fog – it can be used to interpolate and measure the distance between objects under various atmospheric conditions in the development of visual systems for unmanned vehicles.

How a neural network improves photos. Examples of open source solutions

Deep Learning is a relatively new area of Machine Learning, and it can be effectively applied to image processing. Different types of neural networks can be used to solve various tasks to improve image quality, for example, to remove noise, restore high-resolution images from low-resolution image data (super-resolution).

AI-based automatic image processing tools using deep learning algorithms can help you develop intelligent applications, implement computer vision elements, detect and recognize objects and actions in images and videos.

Although different algorithms give different image quality, interpolation usually improves the image in low resolution. But the attempt to “finish” the missing pixels in most cases catches the eye. As a result, the image occupies the entire 4K screen, but it may look blurry or muted, or have artifacts (noise halos, squares). When increasing the video resolution with the help of artificial intelligence, a completely different approach is used.

When receiving a low-resolution image, a deep learning algorithm (usually a generative neural network) predicts a higher-resolution image, which after compression would look like our original at the input. To achieve this level of accuracy, neural networks must be pre-trained on millions of images. When processing low-resolution video, the AI model can “finish” the lost pixels by analyzing each frame and increasing the quality exponentially. This gives an incredible clarity and detail of the picture that no traditional upscale can recreate: from realistic details to a cleaner and smoother reproduction of movements [1].

AI models are constantly being improved, and their accuracy increases with each training cycle. In addition, different deep learning models can be used to improve different types of video materials. And neural networks can also be combined: sometimes it gives amazing results.

**Background and foreground are relative concepts.** Take the motorway as an example: sometimes we are interested in cars that come and go along the highway. At this time, the car is the foreground, and the road and the environment are the background; Pedestrians on the highway are interested, and the intruder is the foreground, and other things, including cars, become the background. There are many ways of background modeling, both advanced and simple. However, different background models have their own reasons, and even advanced background models cannot be applied to any occasion. Below I will present the background modeling methods that were implemented in OpenCV one by one or presented in the book “Learning OpenCV” [1].

### 1. Frame difference

We can say that the difference in frames is the simplest background model. Specify the image in the video as the background, compare the current frame with the background and filter out small differences if necessary. The result is the foreground. OpenCV provides us with a function for dynamically calculating the threshold and then using the frame difference for the foreground detection-cvchangedetection (Note: EmguCV does not encapsulate cvChangeDetection, I declare it to the OpenCvInvoke class, see the code at the end of the article for a specific implementation).

```

Their implementation code looks like this: [DllImport("cvaux200.dll")] public
static extern void cvChangeDetection(IntPtr prev_frame, IntPtr curr_frame, IntPtr
change_mask); // backgroundMask – это фон, imageBackgroundModel – это
фондовая модель, а currentFrame - текущий кадр. If (backgroundMask == null)
background Mask = new Image<Gray, byte>(imageBackgroundModel.Size); if
(threshold == 0d) //If the threshold is 0, use adaptive dynamic background detection
in OpenCv OpenCvInvoke.cvChangeDetection(imageBackgroundModel.Pt,
currentFrame.Ptr, backgroundMask.Ptr); else { // If a threshold is set, use the frame
differenceImage<TColor, Byte> imageTemp = imageBackgroundModel.Abs
Diff(currentFrame); Image<Gray, Byte>[] Images = imageTemp.Split(); background
Mask.SetValue(0d); foreach (Image<Gray, Byte> image in images) background
Mask._Or(image.ThresholdBinary(new Gray(threshold), new Gray(255d))); }
backgroundMask._Not();

```

In cases such as unattended theft from a warehouse, the effect of using the frame difference is very good.

### 2. Background statistical model

A background statistical model is: background statistics over a period of time, and then calculates its statistics (such as mean, mean difference, standard deviation, mean drift, etc.) using statistics as a background method. OpenCV does not implement a simple background statistical model, but the average background statistical model in “Learning OpenCV” has a very detailed introduction. Based on the simulation of the algorithm, implemented a number of background statistical models, including: mean background, mean offset, standard deviation and standard covariance. [2]

Calling the background statistical model is very simple, just 4 steps:

```

//(1) Initialize an object BackgroundStatModelBase<Bgr> bgModel = new
BackgroundStatModelBase<Bgr>(BackgroundStatModelType.AccAvg);//(2) Update

```

the background image for a certain period of time and call it again according to the situation (2)bgModel.Update(image);/(3) Set the current framebgModel.Current Frame = currentFrame;/(4) Get background or foreground Image<Gray,Byte> imageForeground = bgModel.ForegroundMask;

### 3. The code book of the background model.

The main idea of the codebook is as follows: for each pixel change on the time axis, create several (or one) blocks (ranges of variables) that take into account all recent changes, if detected, use the current pixels to compare with the block. If the current pixel falls within the range of any block, it is the background.

The codebook background model has been implemented in OpenCV, but the implementation method is slightly different from the method mentioned in “Learning OpenCV”, mainly includes: (1) using a singly linked list to place a code element; (2) clearing negative code Element, not reset.

cvCreateBGCodeBookModel Create Background Model

cvBGCodeBookUpdate Update Background Model

cvBGCodeBookClearStale eliminates negative code element

cvBGCodeBookDiff calculates background and foreground (note: this function only sets the background pixel to 0, and the foreground pixel is not processed, so you need to set all pixels to the foreground before calling) cvReleaseBGCode BookModel frees up resources.

### REFERENCES:

1. Bezrukov B.N. Specification of video monitoring of broadcast television images, Materials of the HAT International Congress, Moscow, 2002. – С. 215–216.
2. Vorobel R.A. Image contrast improvement using a modified method of lump stretching. Selection and processing of information / R.A. Vorobel, I.M. Journal – M.: 2000, – № 14 (90), – С. 116–121.
3. Gonzalez R., Woods R. Digital image processing / Pereyev. from English – M.: Technosphere, 2006. – 1070.
4. Gonzalez R. “Digital image processing” [Text] / R. Gonzalez, R. Woods – M.: “Technosphere”, 2005. – 1073.