Method for Visual Video Defects Detection using Machine Learning

Dmytro Fedasyuk[a], Roman Lukomskyi[a], Tetyana Marusenkova[a]

[a]*Lviv Polytechnic National University, St. Bandery str, 28 a, Lviv, 79013, Ukraine*

Abstract

The main problem to be solved by this research is the imperfection of the video testing process. Nowadays this process involves mainly manual testing, which is inefficient due to the high probability of human errors, significant time, and material costs. In order to improve this process, we have created a convolutional neural network that can detect defects in video frames with high probability. We have also built a prototype of a software system that can automatically detect defects in video using the created and trained convolutional neural network.

Keywords

Defects detection, image recognition, machine learning, deep learning, convolutional neural networks, automation, video testing.

1. Introduction

The evolution of digital communication systems has led to the active development of multimedia systems and applications such as IPTV (Internet protocol television), mobile multimedia, social networks, virtual reality games, video conferencing, and educational multimedia presentations. These multimedia applications have now become an integral part of everyday life, and they are expected to grow rapidly in the future. Video is being widely used in the above-mentioned application areas, which set strict requirements to its quality, i.e., video content should be free of defects.

A visual defect or a perceptual artifact in a video is a noticeable frame distortion. Such distortion can appear as a result of errors in compression, transmission, or encoding. An example of a visual defect is shown in Figure 1.

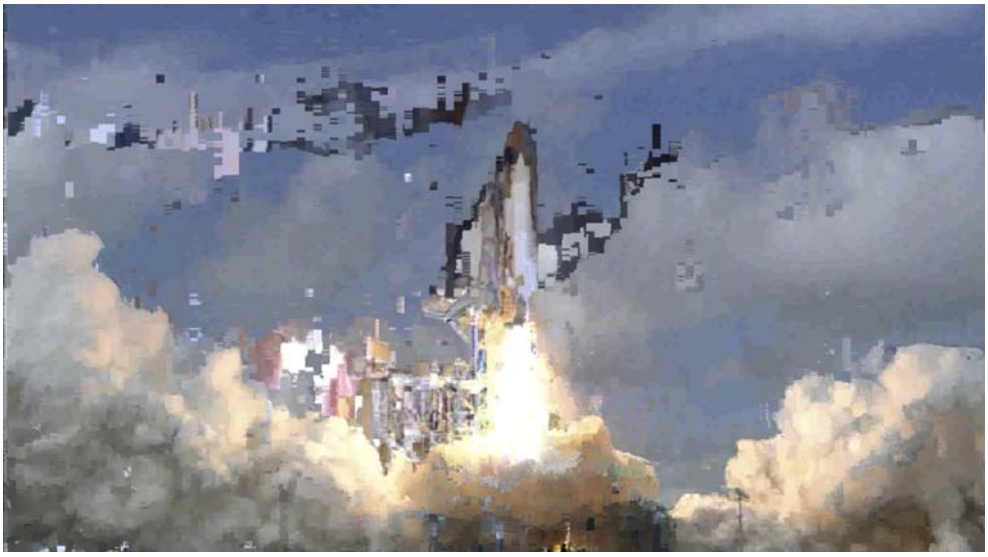

Figure 1: An example of a perceptual artifact

CMIS-2021: The Fourth International Workshop on Computer Modeling and Intelligent Systems, April, 27, 2021, Zaporizhzhia, Ukraine
EMAIL: dmytro.v.fedasyuk@lpnu.ua (D. Fedasyuk); roman.lukomskyi.mnpz.2019@lpnu.ua (R. Lukomskyi); tetiana.a.marusenkova@lpnu.ua(T. Marusenkova)
ORCID: 0000-0003-3552-7454 (D.Fedasyuk); 0000-0002-0345-8290 (R. Lukomskyi); 0000-0003-4508-5725 (T. Marusenkova)

Even minor perceptual artifacts can have a significant impact on the satisfaction of users from watching videos and using multimedia services. With that in mind, multimedia providers are paying more and more attention to ensure the high quality of their video content. Nowadays, the major methods for video quality assessment involve manual testing which is inefficient due to the human factor.

The purpose of this research is to improve the process of video testing. The use of the developed method should reduce the focus on manual testing of video content quality because manual testing has many disadvantages. For example, such disadvantages include the relatively high cost of this process due to the significant cost of viewing equipment, premises, etc. Another important drawback is the high probability that a certain defect will be missed by testers because in order to save time, they usually review only small test fragments, which may not have a particular visual defect in them.

It is expected that the use of the method will increase the percentage of detected defects during testing and, accordingly, will help to improve the overall quality of video and service. An important advantage is that the input of the method is only the video that is being tested. This method, unlike some existing methods of objective quality determination, will allow usage of the software system in cases where the original video file is not available. This is an important factor because in real conditions there is often no access to the original video.

Given the rapid development of the video services market, we can conclude that the development of a method for detecting visual defects is quite promising. The presence of even minor and short defects can significantly affect the satisfaction of users, so companies that provide services related to video distribution and viewing will be interested in improving their video testing process.

The object of research is the process of testing video with the detection of visual defects. The subject of research is methods of detecting defects in the video.

2. Related work

Visual quality assessment refers to gauging a probability of a visual perceptual artifact. Since humans are the ultimate consumers of video content, the subjective methods of testing video quality require manual testing, i.e. involving people who view fragments of video and give a probably biased assessment of its quality on a particular rating scale [1, 2]. Because video content is very large, testers usually are not able to view it entirely.

There are various methods of subjective video quality assessment which determine the rules for selecting an optimal test video fragment duration, number of people that will conduct the assessment as well as metrics that should be used to get the final result. For example, there is a recommendation that suggests that an optimal test fragment length is 10 seconds [3]. However, studies have been conducted to optimize this time indicator [4]. These studies claim that in some cases, reducing the duration of test fragments may not have a significant impact on the resulting testing, but at the same time significantly reduce the time spent on testing. This is evidenced by previous studies that show that, firstly, testers become less attentive if the fragment length exceeds 10 seconds [5], secondly, shorter test sequences contribute to more consistent results [6] and, thirdly, the average shooting time in most modern films is less than 10 seconds [7]. However, when optimizing test suites, there are risks associated with the possibility that defects will be contained in fragments that have been removed from the test suite. Besides, there are various laboratory factors, including, for example, screen size, lighting, viewer-to-screen distance, and so on [8].

Subjective video testing provides reliable results in some cases since it is conducted by humans. On the other hand, this approach has a number of drawbacks for the same reason. To name a few, they include but not limited to:

- the need for significant human resources;
- significant costs for the viewing equipment;
- it takes a lot of time;
- there is a possibility that a certain fragment that actually contains a defect will not be selected to the test fragments set or a short defect will go unnoticed by the tester.

Moreover, this approach cannot be used in continuous quality assessment systems. That is, if a new file is added to the video file database, it requires a separate testing session, which often cannot

take place immediately after receiving the file, but must be scheduled for the future due to the human factor. This also contributes to the risk that a video file will be skipped.

Another significant disadvantage is that very often the same video file is encoded with different quality parameters for adaptive transmission, which greatly increases the number of test videos and, accordingly, the resources spent on testing. Currently, there are methods of objective video quality testing which are being developed in order to solve these problems. They are often referred to as Objective Quality of Experience (QoE) methods.

The structural information of the video is especially important for the comprehension of a person's visual perception. There is an approach to image quality assessment based on structural information, which determines the degradation of structural similarity derived from the statistical properties of local information between the reference and the distorted images [9]. This approach has also been proposed for video. In addition, a criterion of information reliability was proposed to assess the quality of images [10], which is based on statistical characteristics of the structural information of natural images.

Image motion information is also important for optimal, realistic quality assessment. Various researchers have proposed methods for determining quality based on motion information [11, 12]. In particular, in the Motion-based Video Integrity Evaluation (MOVIE) [13], proposed for natural video, distorted and reference content is decomposed using Gabor spatio-temporal filters, therefore the quality index consists of two components – spatial and temporal.

The spatio-temporal quality method provides a significant improvement in performance in terms of statistical correlation between the results obtained with this objective method and the subjective data obtained from humans. The authors note that this method can be used mainly for videos with natural content. The authors also acknowledge that the method is not computationally efficient. This is due to the fact that, since this method involves the decomposition of a video using a large number of Gabor filters, which, in turn, require a large number of frames, the use of the method requires the availability of significant computing resources. Taking this into account, we can conclude that the possibilities for using this method are limited.

Despite some progress in the objective detection of defects in the video, there are a number of issues that need to be addressed [14]. In particular, it is difficult to use the existing objective models in general cases, as each has its own characteristics, which are associated with certain types of content, context, or defects. Because of this, a particular model may do well with some types of video, but have very poor results, in cases of use for another video because it was not configured for it, or this video does not have the statistical characteristics on which the model is built.

Given the different nature of multimedia data, the creation of a generalized model, the logic of which will not depend on the above factors will significantly improve the state of modern video quality assessment.

In addition, traditional methods of quality assessment are often based on explicit modeling of human perception. As a result, systems based on these methods are prone to so-called overfitting and therefore have questionable results on real data. Instead, machine learning-based methods could mimic human perception of quality, rather than developing a precise model of the human visual perception system. Moreover, such methods will not need the original media file in order to assess quality.

3. Using the convolutional neural networks

3.1. The neural network architecture

In order to check if visual defects are present in the video, we can divide it into frames and analyze them separately. The use of such an approach gives us an ability to get detailed results that not only show if the defect is present or not but also, in case if it is present, it can indicate the time when the defect occurred. This might be especially handy if there is a need for a human to double-check if there is a real defect before marking the video as defective.

We can use artificial neural networks (ANN) in order to detect visual defects in specific frames. Artificial neural networks are software implementation of the logic of brain structures. We can train them by specifying input information and examples of relevant output information [15].

As a matter of fact, the biggest limitation of artificial neural networks is that they inevitably require a big amount of computational resources when working with a large amount of input data. This might not be an issue for small black and white images, but the image data in real-world videos always contains a lot of input information (taking into account its size and color) which renders traditional neural networks useless when applied in image recognition tasks. In order to resolve this problem, there needs to be an optimization of the input data that is passed to the fully connected layers. We need to separate image features that are truly important in the given recognition task.

Convolutional Neural Networks (CNN) are a class of artificial neural networks for deep learning that is often used in visual image analysis [16]. The architectural structure of a conventional neural network is fairly simple – each connection has its weight which is used during the process of propagation. The convolutional neural networks try to optimize their input before passing it to a fully connected layer. When it receives large input data (such as a colored image), it uses a number of convolution kernel matrices to retrieve important information from the given data. The important data in the case of a graphical input can be relative locations of certain shapes such as circles, arcs, and lines, distribution of color across the image, etc. When a fully connected layer receives information about the presence or absence of these features as well as their relative location information, it can effectively conduct the image recognition process since the volume of this input is tremendously smaller compared to the full input. A basic CNN architecture is shown in Figure 2. Usually, a CNN architecture would contain several convolutional and pooling layers.

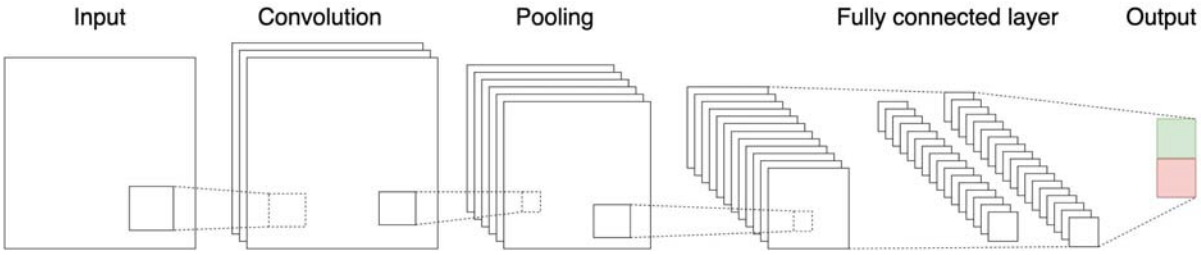

Figure 2: Basic CNN architecture

Naturally, in a convolutional neural network, a set of weights encodes important visual features of the image (color distribution, angles, lines as well as their relative location information). The network's convolution kernels are formed during the training process [17]. They cannot be set in advance, since each image recognition task has its own important features (recognition of certain objects would require a different set of graphical features than recognition of defects).

The pooling operation reduces the size of the formed feature maps. In this network architecture, it is considered that information about the presence of the specific feature is more important than accurate knowledge of its coordinates, so the maximum of several neighboring neurons of the feature map is selected and taken as one neuron of a compact feature map of smaller size [16]. Because of this operation, in addition to accelerating further computations, the neural network becomes more adaptable to the scale of the input image.

A convolutional neural network consists of input and output layers, as well as several hidden layers. Hidden CNN layers typically consist of convolutional, pooling, fully connected, and normalization layers. Convolutional layers apply a convolution operation to the input data, passing the result to the next layer. The convolution operation simulates the response of a single neuron to a specific visual stimulus. Each convolutional neuron processes data for its receptive field. The convolution operation solves the problem of the increasing number of connections, as it reduces the number of parameters, allowing the network to be deeper [16]. The convolutional layer is the main building block of CNN. The parameters of this layer consist of a set of filters for learning, which have a small receptive field. During the forward propagation, each filter performs a convolution on the width and height of the input layer, calculating the scalar product of the filter and input data, and forming a 2-dimensional activation map of this filter. As a result, the network learns which filters are activated when it detects a particular type of feature in a particular spatial position in the input data.

When processing multidimensional inputs (such as images), it is impractical to connect all neurons with all neurons of the previous layer, because such a network architecture does not take into

account the spatial structure of the data. Convolutional networks use spatial-local correlation by providing a scheme of local connection between neurons of adjacent layers: each neuron is connected to only a small area of the input layer. The receptive field of a neuron is a hyperparameter that is the degree to which neurons connect. Connections are local in space (along width and height) but always propagate along the entire depth of the input layer.

The spatial size of the output volume is calculated using a formula (1).

$$N = (W - F + 2P)/2 + 1, \tag{1}$$

where N – the output volume size,
W – the input volume size,
F – the kernel size,
P – the amount of zero padding,
S – the stride.

In order to ensure that the dimensions of the input and output matrices are equal (provided that the step of the filter area is equal to one), the size of the zero padding is determined by the formula (2).

$$P = (F - 1)/2, \tag{2}$$

where P – the amount of zero padding,
F – the kernel size.

Convolutional and pooling layers are followed by fully connected layers (Figure 3). Neurons in the fully connected layer are connected to all neurons in the previous layer, just as they do in conventional artificial neural networks [18].

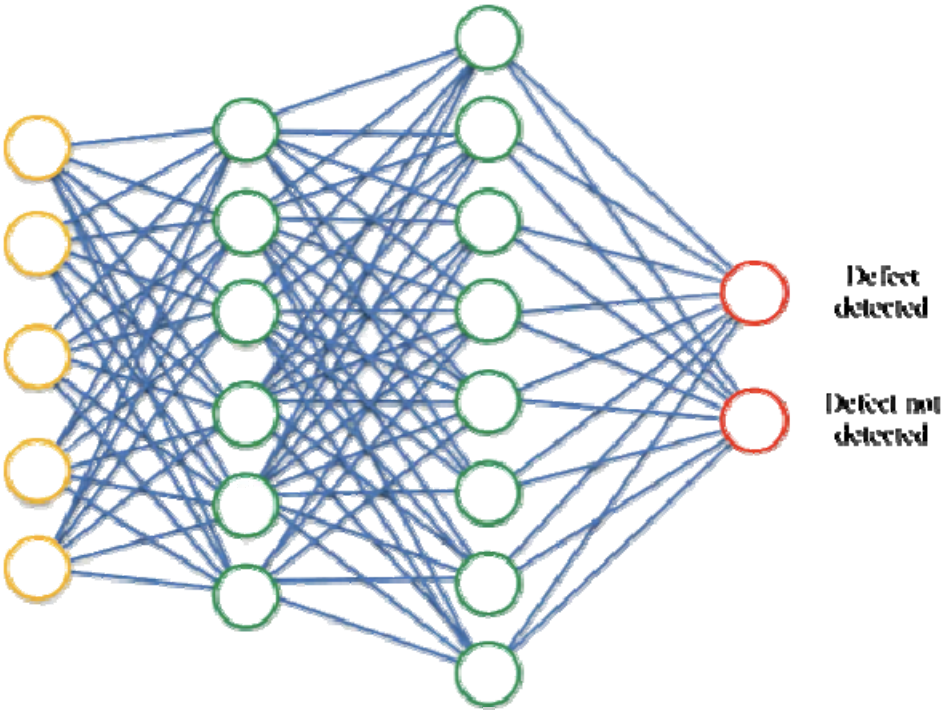

Figure 3: Fully connected layers

The loss layer determines how the training process penalizes the deviation between the predicted and expected results and is usually the final layer. It can use different loss functions for different tasks. For example, normalized exponential (softmax).

Because the fully connected layer has the most parameters, it is prone to overfitting. One of the most common methods of reducing overfitting is a dropout. At each stage of training, individual nodes with a certain probability are either "excluded" from the network or remain in it, thus, as a result, the network is reduced. Inbound and outbound links of excluded nodes are also deleted. In the

next step, only a reduced version of the neural network is trained on the data. The removed nodes are then re-inserted into the network with their previous weights.

Taking into account the above facts it was decided to choose the neural network architecture which is shown in Figure 4.

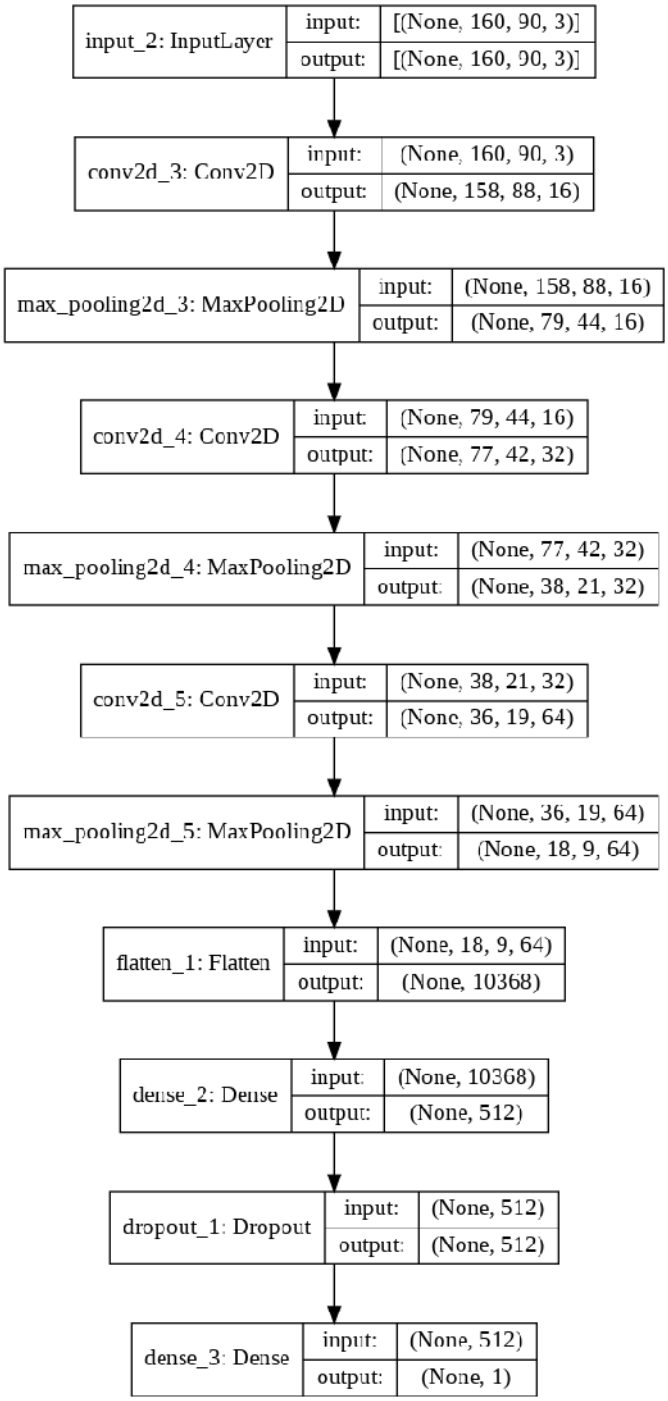

Figure 4: Architecture of the created convolutional neural network

3.2. Dataset construction

In order to construct the dataset, we selected several creative commons licensed videos from the Internet. To provide the best learning results during the neural network training process, the dataset needs to have a wide variety of training data. In order to achieve this, we made sure that the selected video content includes a diverse set of content genres. In addition to that, the scenes in the videos include different types of camera motion such as static, moving, and zoom. Additionally, we

have ensured that the videos are of pristine quality by carefully inspecting each of them. Making sure that the training data is labeled correctly will help to avoid confusing the convolutional neural network during the training process.

In order to have distorted versions of the same videos, we have simulated errors in the original videos. After that, we have once again ensured that the distorted videos indeed have visual defects in each frame. Later, we broke down the videos into separate frames so that they could be given to the CNN as its input.

The final dataset consists of 129,092 video frames. Each frame from the dataset is labeled as either “damaged” or “original” as shown in Figure 5. For each damaged frame there is an original frame in the dataset and vise versa. This way the convolutional neural network can learn most effectively since it has similar reference images for both classes.

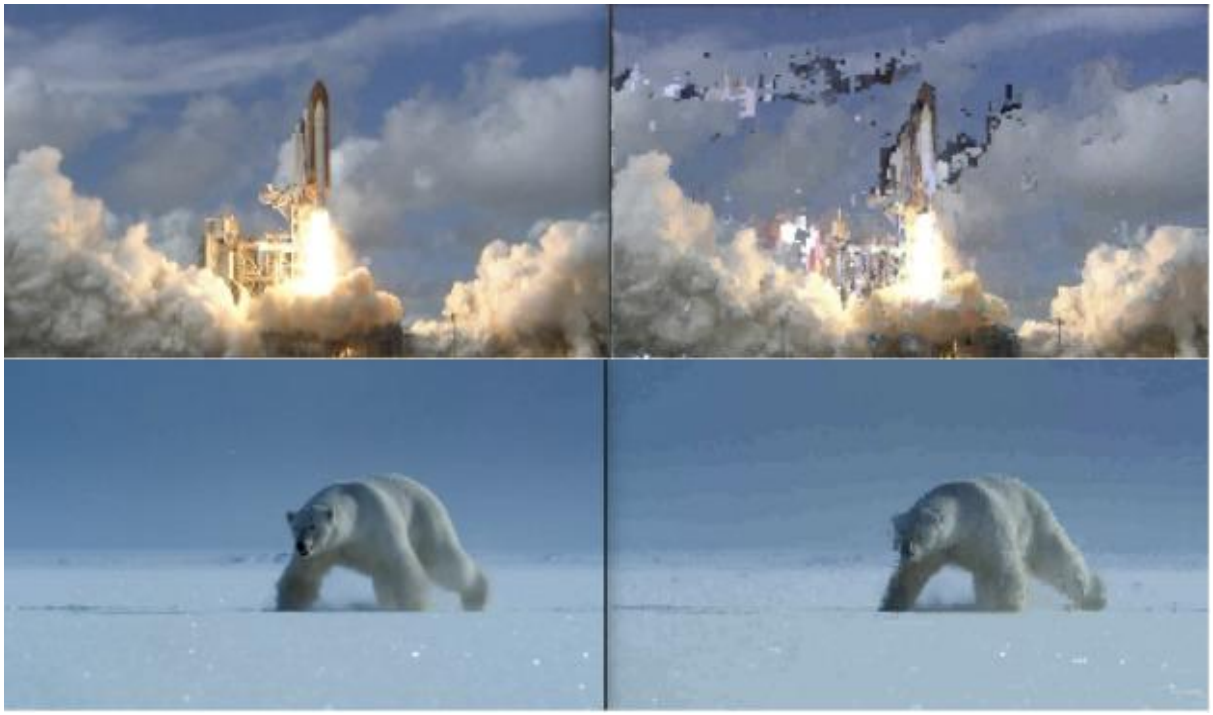

Figure 5: Original (on the left) and distorted (on the right) frames from the dataset

3.3. Training

Since convolutional neural networks require a lot of training data to perform well, it is common to use the existing entries in the dataset in order to generate new training examples. This technique is called data augmentation [18, 19]. Advanced data augmentation might include the usage of generative adversarial network (GAN) [20] or balancing GAN (BAGAN) [21]. However, taking into account the nature of our data, especially the importance to keep all of our training examples as close to natural samples as possible, these techniques are not needed in our case. Traditional transformations consist of using a combination of transformations on the original data. Since we are working with images, such transformations might include:

- horizontal/vertical image flip;
- rotation;
- cropping;
- translation;
- shift;
- zoom;
- shading.

Experimentally we have discovered that in the case of visual defects detection, image flip performs the best out of all of the above approaches. This might be explained by the fact that other of the above techniques might introduce distortions in the dataset images marked as “original” (for

example, excessive pixelation after zooming in or out). This would have a significant negative effect on the training process and the final CNN accuracy. Taking this into account, we have decided to apply the data augmentation using the image flip technique.

In order to conduct the training, we used Tensorflow and Keras with Python. As a result of training on the previously created dataset, the neural network managed to achieve an accuracy of 98.5%. The process of training with the respective changes in both training and validation loss and accuracy is shown in Table 1.

Table 1
The training process

Epoch	Training loss	Training accuracy	Validation loss	Validation accuracy
1	0.2121	0.9146	0.3368	0.9148
2	0.0973	0.9719	0.0710	0.9785
3	0.0845	0.9786	0.0663	0.9835
4	0.0826	0.9806	0.0443	0.9850

As can be seen from Table 1, the created convolutional neural network can detect visual artifacts in frames with decent probability.

We can now use the created convolutional neural network to detect visual defects in the video. Using a trained model, we have built a prototype of a software system that checks for visual defects in the video by cyclically performing the following steps:

- retrieving a video frame – in this step of the algorithm, the prototype of the software system forms a video frame in the final image form – the same form in which it would be shown to the viewer;
- analyzing the video frame – the convolutional neural network analyzes the video frame which was obtained in the previous step. In this step the CNN will output the probability of visual distortion presence;
- saving the result of the analysis – the software system stores the result of the analysis in a particular video frame in order to be able to later show it to the user on the graphical interface;

The described algorithm is shown in Figure 6.

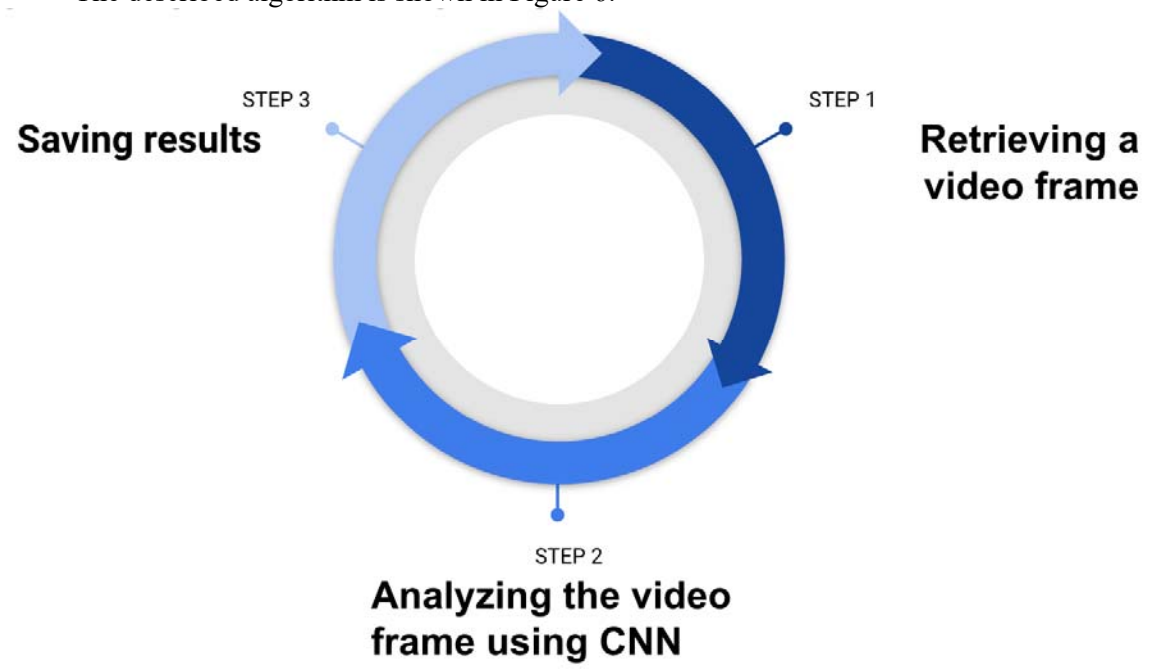

Figure 6: Algorithm of the software system operation

After performing the steps described above, the prototype of a software system builds a diagram of the defects as shown in Figure 7. Normally, at this point, a person who is responsible for the testing process would need to take a look at the diagram and investigate the defects that have been found.

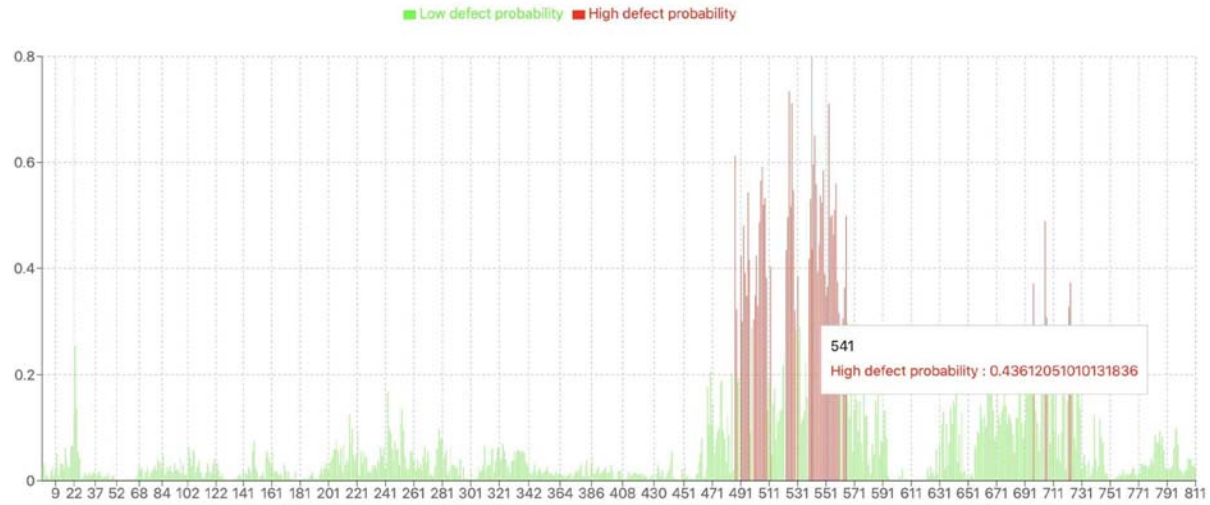

Figure 7: The defects probability diagram

4. Conclusion

In this paper, we analyzed the problem of video testing. The video content in modern video content delivery systems undergoes multiple stages that can introduce visual distortions. Such distortions can have tremendous effects on the end-user experience.

Manual testing takes an unreasonable amount of time and leads to a high probability of human error. In spite of different methods and ideas for improving defect detection, there are a number of issues such as the need for significant human resources, significant costs for the viewing equipment, limited usage for continuous quality assessment systems. To add, traditional methods of quality assessment are often based on explicit modeling of human perception. Therefore, systems based on such an approach are prone to overfitting and have questionable results on real data. Machine learning-based methods could solve this issue. Moreover, such methods will not need the original media file in order to assess quality.

In order to solve this issue, we proposed using a convolutional neural network. Machine learning-based methods can mimic human perception of quality and it can increase the percentage of detected defects during testing. In order to conduct the training of the CNN, we have created a dataset from pristine and distorted videos which consists of 129,092 frames. In order to provide even more training data to the CNN, we used the data augmentation technique. As a result of training, the convolutional neural network achieved an accuracy of 98.5%. Taking into account the high accuracy of the trained model we can see that it is possible to use it in order to detect visual defects in video frames.

The created model was used to build a prototype of a software system that can detect defects in video with quite a high accuracy. The use of the developed software system can reduce the human factor in the process of video testing, as well as significantly speed up and reduce the cost of this process. Saving the result of the analysis and diagram of the defects helps the user to easily analyze the detected defects.

5. References

[1] A. K. Moorthy, K. Seshadrinathan, A. C. Bovik, Image and Video Quality Assessment: Perception, Psychophysical Models, and Algorithms, Perceptual Digital Imaging: Methods and Applications (2017) 55-81.

[2] M. H. Pinson, S. Wolf, Comparing subjective video quality testing methodologies, Communications and Image Processing (2003) 573-582. doi:10.1117/12.509908.

[3] ITU-R, Methodology for the Subjective Assessment of the Quality of Television Pictures, 2002. URL: http://www.gpds.ene.unb.br/databases/2012-UNB-Varium-Exp/Exp3-Delft/00-report-alexandre/Papers---Judith/Subjective%20Studies/ITU-Recommendation---BT500-11.pdf
[4] F. M. Moss, K. Wang, F. Zhang, R. Baddeley, D. R. Bull, Moss, Felix Mercer, Ke Wang, Fan Zhang, Roland Baddeley, and David R. Bull. "On the optimal presentation duration for subjective video quality assessment, IEEE Transactions on Circuits and Systems for Video Technology (2015) 1977-1987.
[5] P. Fröhlich, S. Egger, R. Schatz, M. Mühlegger, K. Masuch, and D. Gardlo, QoE in 10 seconds: Are short video clip lengths sufficient for quality of experience assessment?, 2012 fourth international workshop on quality of multimedia experience. IEEE (2012) 242-247.
[6] B. W. Tatler, R. J. Baddeley, and I. D. Gilchrist, Visual correlates of fixation selection: Effects of scale and time, Vision research 45, no. 5 (2005) 643-659.
[7] J. E. Cutting, K. L. Brunick, J. E. DeLong, C. Iricinschi, A. Candan, Quicker, faster, darker: Changes in Hollywood film over 75 years, i-Perception 2 (2011) 596-576. doi:10.1068/i0441aap.
[8] Q. Huynh-Thu, M. Ghanbari, D. Hands, M. Brotherton, Subjective video quality evaluation for multimedia applications, Human Vision and Electronic Imaging XI, vol. 6057, p. 60571D. International Society for Optics and Photonics (2006). doi: 10.1117/12.641703.
[9] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE transactions on image processing 13(4) (2004) 600-612. doi:10.1109/TIP.2003.819861.
[10] H. R. Sheikh, A. C. Bovik, G. De Veciana, An information fidelity criterion for image quality assessment using natural scene statistics, IEEE Transactions on image processing 14, no. 12 (2005) 2117-2128. doi:10.1109/TIP.2005.859389.
[11] K. Seshadrinathan, A. C. Bovik, Motion tuned spatio-temporal quality assessment of natural videos, IEEE transactions on image processing 19.2 (2009) 335-350. doi:10.1109/TIP.2009.2034992.
[12] R. Soundararajan, A. C. Bovik, Video quality assessment by reduced reference spatio-temporal entropic differencing, IEEE Transactions on Circuits and Systems for Video Technology (2013) 684-694. doi:10.1109/TCSVT.2012.2214933.
[13] K. Seshadrinathan, A. C. Bovik, Motion-based perceptual quality assessment of video, Human Vision and Electronic Imaging XIV (2009). doi:10.1117/12.811817.
[14] Z. Akhtar, T. H. Falk, Audio-Visual Multimedia Quality Assessment: A Comprehensive Survey, IEEE Access (2017). doi:10.1109/ACCESS.2017.2750918.
[15] C. Aggarwal, Neural Networks and Deep Learning: A Textbook, Springer, New York, NY, 2018.
[16] K. O'Shea, R. Nash, An Introduction to Convolutional Neural Networks, arXiv preprint arXiv:1511.08458 (2015).
[17] F. Millstein, Convolutional Neural Networks In Python: Beginner's Guide To Convolutional Neural Networks In Python, Scotts Valley New York, NY, 2018.
[18] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, arXiv preprint arXiv:1712.04621 (2017).
[19] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, Journal of Big Data 6.1 (2019) 1-48. doi:10.1186/s40537-019-0197-0.
[20] I. Goodfellow, Nips 2016 tutorial: Generative adversarial networks, arXiv preprint arXiv:1701.00160 (2016).
[21] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, C. Malossi, Bagan: Data augmentation with balancing gan, arXiv preprint arXiv:1803.09655 (2018).