Frames Preprocessing Methods for Chromakey Classification in Video

Evgeny Bessonnitsyn, Artyom Chebykin, Grigorii Stafeev and Valeria Efimova[a]
ITMO University, Kronverksky Pr. 49, bldg. A, St. Petersburg, 197101, Russia

Keywords: Chromakey, Transformer, Video Analysis, Classification, Deep Learning, Image Processing.

Abstract: Currently, video games, movies, commercials, and television shows are ubiquitous in modern society. However, beneath the surface of their visual variety lies sophisticated technology, which can produce impressive effects. One such technology is chromakey — a method that allows to change the background to any other image or video. Recognizing chromakey technology in video plays a key role in finding fake materials. In this paper, we consider approaches based on deep learning models that allows to recognize chromakey in video based on unnatural artifacts that arise during the transition between frames. The video consists of a sequence of frames, and the the video accuracy can be determined in different ways. If we consider the accuracy frame by frame, our method reaches an F_1 score equal to 0.67. If we consider the entire video to be fake in case there is one or more fake segments, then the F_1 score equal to 0.76. The proposed methods showed better results on the dataset we collected in comparison with existing methods for chromakey detection.

1 INTRODUCTION

Recognizing chromakey technology in video is one of the key innovations in realistic and immersive creation field. The basic technology principle is to use a special color background, usually green or blue, which is easily distinguishable from other objects in the video; less often, white, yellow and other colors are used for a replaceable background. Then, using special software, the background is recognized and separated from objects or people. First, the color of the background to be removed is selected, and then the desired image is substituted in its place.

Modern laptops and smartphones use simpler technologies to create the effect. For them, it is not necessary to use a solid special color background, and the objects separation from the background occurs using neural networks. However, this approach has significantly lower accuracy than classic chromakey. For example, when an object moves, a “halo” may appear with the previous background parts.

In general, if chromakey is used unprofessional, artifacts may arise in the form of overly clear or, conversely, unnatural blurry contours. Moreover, chromakey artifacts include a sharp difference in the foreground object lighting from the background lighting. By the such artifacts quantity, we can determine whether the video is fake or real. However, after qualitative post-processing stage, it is quite difficult for the human eye to recognize negative effects, and one has to draw conclusions based on the video plot realism degree.

Moreover, video frames classification without preprocessing leads to the fact that naive algorithms, like the human eye, do not notice the difference between fake content and real content. Therefore, in computer vision methods it will not be possible to use pretrained models, since they do not take into account the preprocessing clues.

In the past, due to these problems, analytical methods and attempts to preprocess data for further analysis were used to solve such problems. In this paper we develop preprocessing approach and combine it with deep learning methods.

Our research is aimed primarily at finding optimal approaches to data preprocessing and their further implementation into classical methods. Due to the reasons described above, at the moment there is no large and high-quality dataset for chromakey recognition. Our tasks also included its formation based on short videos.

Our contribution is threefold:

- We propose frame preprocesing, which can be used with image classification networks.
- We proposed frame preprocesing, which can be used with vision transformers.
- We release the labeled dataset of 200 video links with chromakey.

We provide access to the code and data set [1].

[a] https://orcid.org/0000-7981-878-1424

[1] https://github.com/Turukmokto/Chromakey

Bessonnitsyn, E., Chebykin, A., Stafeev, G. and Efimova, V.
Frames Preprocessing Methods for Chromakey Classification in Video.
DOI: 10.5220/0012467900003660
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 3: VISAPP*, pages 227-232
ISBN: 978-989-758-679-8; ISSN: 2184-4321
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

In Section 2 we will talk about existing approaches to solving the problem. In Section 3 we will describe the method we developed using machine learning models. Finally, in Section 4 we will demonstrate our approach results.

2 RELATED WORK

2.1 Analytical Method

So far, the analytical approach for high-quality video preprocessing played a key role in chromakey recognition (Singh and Singh, 2022). This paper highlights the fact that chromakey is much easier to recognize in motion, which means that it is necessary to evaluate not individual frames, but the pixel-by-pixel difference between them. At the final processing stage, the difference frame is converted to grayscale. Since chromakey leaves artifacts or unnatural differences in lighting mainly at the objects edges, thus, on the difference frame should be left only pixels related to objects contours obtained by Canny edge algorithm (Ding and Goshtasby, 2001).

$$Fr_{diff} = |Fr_i - Fr_{i+1}|, \tag{1}$$

$$Sample_i = GrayScale(CannyEdge(Fr_{diff})), \tag{2}$$

where Fr_i is i'th frame in video sequence.

After preprocessing, the paper suggests an analytical method for assessing such a difference frame using a threshold value obtained from image binarization using the Otsu algorithm (Otsu, 1979). Unfortunately, this approach is ineffective on video with high-quality post-processing and a small object movement amount, since generally it is not possible to analytically evaluate most of the edge cases. It follows that the current approach is imperfect and requires significant improvement and generalization to existing methods of chromakey creation.

2.2 Computer Vision Algorithms

Today, there are many machine learning models that allows solving the classification problem extracting visual features. Among the most famous and effective are EfficientNet (Koonce and Koonce, 2021), ResNet (Targ et al., 2016), Unet (Huang et al., 2020), Vit (Yuan et al., 2021), Swin (Liu et al., 2021) and others. These models can be used for processing RGB images in quite complex and specific tasks, such as identifying diseases (Li et al., 2020), defect classification (Song et al., 2019) and others. With the proper preprocessing of the input frames, it is possible to collect a dataset for training the model that will determine the chromakey technology usage — for example, using the same difference frames principle as a basis.

2.3 Transformers

The Vision Transformer (ViT) model (Yuan et al., 2021) made a huge impression on the computer vision community; it is not only capable to process image as a sequence, but had a higher modeling ability thanks to a new attention mechanism (multi-headed self-attention), designed specifically for transformers for natural language processing tasks, wide receptive fields, as well as smaller biases. All this allowed ViT to achieve the best results (SOTA) in the image classification task on the ImageNet dataset (Deng et al., 2009). Gradually, the transformer architecture began to displace advanced methods in other computer vision areas, including video classification.

The paper about TimeSformer (Bertasius et al., 2021), proposes a method for processing video sequences based on ViT. Embedding for a sequence is created by combining the batch size, the sequence length, dividing each frame into patches, pulling them into vectors, which are then multiplied by the training matrix, and then positional embedding is added to the result. Paper also propose a more efficient spatio-temporal attention architecture, called "Divided Space-Time Attention", in which temporal and spatial attention are divided into two separate blocks and applied one after another.

The paper proposes the use of ViViT (Arnab et al., 2021) model and several alternative approaches to solve the video classification problem. The first method is to create two independent encoders, the first for the spatial embeddings and the second for the spatial encoder outputs to evaluate the entire sequence, this is called a temporal encoder. This approach is very similar to a convolutional neural network: at first the network sees only individual small pieces, but as it moves deeper into the network, the receptive field increases until it becomes equal to the whole image, and then it is fed to the classifier. The second method is to leave one encoder, but divide the attention mechanism into two parts: spatial and temporal. This approach was also described in the paper about TimeSformer. The last method is to create two independent heads groups in Multi-Head Dot Product Attention: spatial and temporal. The last two approaches allows preserving the one method functionality and reducing the calculations complexity. Moreover, the paper provides techniques for initializing model weights for video classification using analogues for image classification, since to train a model from scratch you need a huge video dataset, which is difficult to collect.

The VideoMAE paper (Tong et al., 2022), proposes a solution to the data lack problems, high correlation between frames. Frames sequence is taken, than, divide these frames into patches, randomly apply a tube masking on them with an extremely high coefficient, then create an embedding from this. The embedding is then encoded and decoded trying to recreate the missing patches. This should help the transformer "pay more attention" to dynamic objects, and also not retrain and remember low-level features, but focus on high-level ones.

3 METHOD

In general, there is a problem with the analytical approach to implementing the chromakey recognition algorithm of using this technology in different cases, leading to a large false positive results number. We discuss existing method (Singh and Singh, 2022) problems in Subsection 3.1, after that we propose two chromakey classification methods, the first based on a convolutional neural network (Subsection 3.2), and the second based on the transformer model (Subsection 3.3).

3.1 Existing Problems

The initial idea was to simply replace the analytical approach with a classification algorithm and select the best model. The fact is that despite the good results during training, the results on the test set was much worse. This happens because the model, evaluating one difference frame during the training process, begins to rely solely on the video plot and on the pixels configuration, and not on their unnaturalness relative to others. The model just remembers "scenarios", considering some to be natural and others not. Due to the fact that it is impossible to include every possible video content variety in finite data set, the dataset turns out to be very limited. Consequently, the model performs much worse on the test sample than during the training process.

3.2 CNN-Based Method

Based on the experiment results described above, we hypothesized that a single difference frame may not be enough to detect a fake. Note that our difference frame after preprocessing is a single-channel image, while most machine learning models are made to analyze three-channel RGB images. Thus, it is possible to analyze not two consecutive frames, but four at once, since from them you can make three consecutive single-channel difference frames, and then combine them into one RGB image.

$$Triplet_i = [Sample_i, Sample_{i+1}, Sample_{i+2}], \quad (3)$$

where $Sample_i$ is calculated following equation 2.

As a result, it will describe the change in the image contours over 4 frames, with each frame being used exactly once for each color (RGB). It should be noted that the machine learning model has no information about the image coloring; this is a convention only for understanding the classified images preprocessing. This approach gave better results on the test set, since it assessed a larger video period and could work with previously unknown scenarios. There are three examples of classified samples in Fig 1.

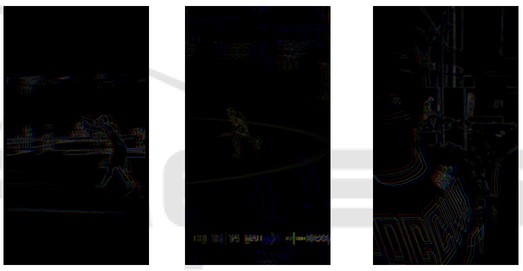
Figure 1: Triplets example.

3.3 Transformer-Based Method

As is the case with computer vision models, transformers do not work well with raw data and "notice" chromakey where there is none in reality. Here we propose a second method for preprocessing the input video. Each frame in a video containing chromakey can be conditionally divided into two parts according to the noise pattern. This is due to the fact that we insert one image into another, that is, the areas have different intrinsic noise (with different characteristics) due to the imperfection of the physical photons converting processes into electrical energy and illumination. All this affects the level and noise nature in the image.

In this method, we will use the idea described above to preprocess the input videos by creating standard deviation maps for each frame. We first convert the image from the RGB format to the YCbCr format, since the RGB color space is inefficient for storing and transmitting color signals (Kaur and Kranthi, 2012). In YCbCr there is no division into primary colors; all of them are converted into visually significant information. We take only the Y channel, since it is responsible for the pixel brightness value, and calculate it using following equation:

$$Y = wr \cdot R + (1 - wb - wr) \cdot G + wb \cdot B, \quad (4)$$

where $wr = 0.299$ and $wb = 0.114$.

Next, we apply a 2D discrete wavelet brightness transform (db4) (Zhang and Zhang, 2019) to the image in order to calculate the noise heterogeneity, because the wavelet coefficients at the first level are, in fact, pure noise.

$$y[n] = (x \cdot g)[n] = \sum_{k=-\inf}^{\inf} x[k]g[n-k], \quad (5)$$

where n — a signal and g — low-pass filter with impulse response.

We are only interested in the diagonal component, since it is an approximation coefficient. Next, we calculate the absolute median deviation (MAD) to find the noise variation, that is, we divide the image into *NxN* blocks and take the median over the blocks.

Basically, near objects with many contours, the noise level will always be slightly higher than where there are no objects. To eliminate this factor, we calculate the minimum for each block and subtract the MAD value for the block elementwise. Finally, we reassemble the image from the blocks. Fig 2 shows the frame preparation process.

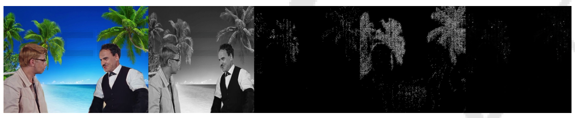
Figure 2: Preprocessing example. At the first step, the original image is converted into the YCbCr format and the preprocessing described above is applied. In the second step, the 2D discrete wavelet brightness transformation is applied. In the third step we calculate the MAD value for each block. In the final step we subtract the MAD value from the second step result.

After processing the frames, we create sequences of the required length from them. Then, due to the lack of a sufficiently large dataset necessary for training the transformer, we decided to feed the transformer with features obtained using a convolutional neural network from a preprocessed frames sequence obtained in the previous step. Intermediate extractor usage also allows us to find very different chromakey appearance on various videos. A schematic diagram of the algorithm is presented in Fig. 3.

Transformer is a DeiT (Xie et al., 2021) encoder that is great for learning from small data amounts. It consists of two layers, containing 4 attention heads. The small size is due to the careful input videos preprocessing presence.

4 EXPERIMENTS AND RESULTS

4.1 Dataset

There is currently no publicly available dataset for classifying chromakey, so we collected it from YouTube. The set contains 200 videos each lasting up to 10 minutes and marked second by second for the chromakey presence or its absence. All videos are 720p quality or higher. In Table 1 are frame statistics for the entire dataset and for the train/test parts.

Table 1: Dataset statistics, # denotes number of corresponding items.

	#total frames	#fake frames	#real frames	videos count
Train	26368	13242	13126	156
Test	115112	61895	53217	44
All	141480	75137	66343	200

In the future, when enlarging or customizing the dataset, it is worth paying attention to several points. For correct testing, videos in the test and training sets should not overlap. In addition, the number of fake and real frames in both samples should be approximately the same.

4.2 Selected Metrics

The algorithms quality can be assessed in three different ways:

- Frame-by-frame evaluation — For each video frame, we try to determine whether the model's results are true.
- Evaluation per second — Each second consists of the frames we measured in the previous metric. To draw a conclusion about the particular second fakeness, it is necessary to select a threshold for the fake frames number in it. Our threshold was selected empirically on the dataset test part. This is a configurable setting, which is manually tuned for specific videos sets. For example, if a dataset consists mainly by static videos, the threshold should be increase to 0.9. If a dataset contains a large number of videos with fast dynamics, it is worth reducing the threshold to 0.5. In our test we used an average value of 0.7.
- Video rating — If a certain number of consecutive fake seconds is found in a video, specified by a custom hyperparameter, then this video can be considered fake. Otherwise, the video is considered real. Based on this metric, you can evaluate the performance on the entire test videos set.

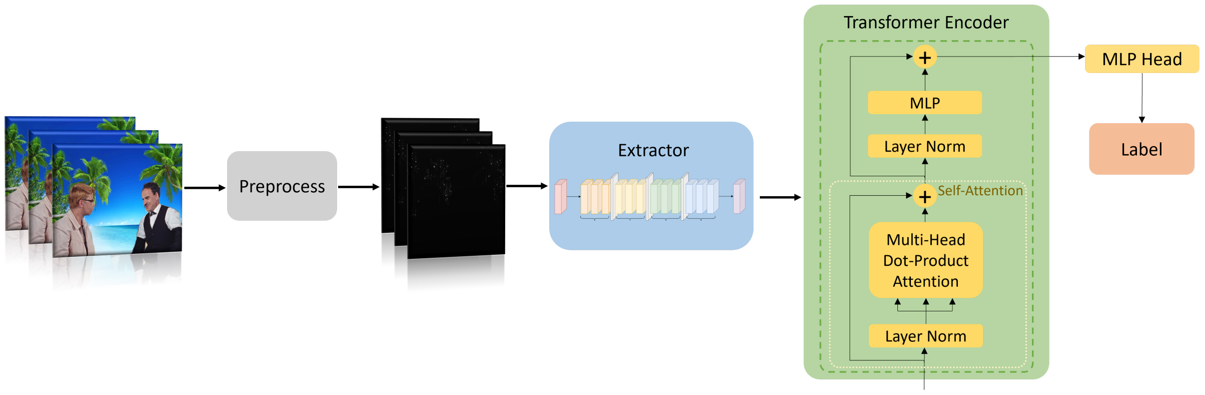

Figure 3: Algorithm scheme.

Table 2: F_1 scores for method comparison, best value is marked with bold.

	Frame-by-frame evaluation	Evaluation per second	Video rating
Analytical approach	0.23	0.27	0.12
EfficientNet B0 w/o preprocessing	0.3	0.31	0.33
EfficientNet B0 with preprocessing	**0.67**	**0.72**	**0.76**
4 frames interval w/o preprocessing	0.4	0.41	0.39
4 frames interval with preprocessing	0.6	0.71	0.75
8 frames interval with preprocessing	0.6	0.56	0.63
16 frames interval with preprocessing	0.61	0.64	0.61
32 frames interval with preprocessing	0.59	0.57	0.58

Since all of these metrics can be important when solving specific problems and allows hyperparameters tuning, we tested our algorithm on all of them. In our testing, we used a second confidence threshold of 0.7, and when evaluating the latter metric, we did not take into account fake intervals less than 3 seconds. The hyperparameters were selected empirically, but can be easily replaced depending on the dataset features.

4.3 Models Results

After the first preprocessing method described in subsection 3.2, we can use any classical computer vision classifier model that work with RGB images such as EfficientNet, ResNet, VGG and others. It is worth mentioning that using pre-trained models gives worse results than those trained from scratch since pretrained models do not take into account data preprocessing. As a result, EfficientNet (B0, B1, B2, B3, B4), ResNet-(50, 101), VGG-16 were trained and tested. After training different versions of these models on our dataset, EfficientNet B0 showed the best results.

For the second preprocessing method described in subsection 3.3, the length of the analyzed video period can be selected. We trained the transformer at intervals of length: $4, 8, 16, 32$ frames. The results evaluated by the metrics described above for proposed methods and analytical approach are presented in the Tab. 2.

Despite the similar results of both methods, we consider the second approach more promising, since the training set size was small, while transformers need more data for training than convolutional neural networks to show superior performance. However even at the current stage, the proposed preprocessing to a large extent compensates for this shortcoming.

4.4 Limitations

Despite the decent results, the methods we propose work differently on various types of video. CNN-based method 3.2 copes much worse with videos in which there is a small number of movements, and as a result there are few contours for analysis. The Transformer-based method 3.3 copes better with static videos, but works much worse on bright videos, since due to excessive brightness it is not possible to identify the different nature of noise in the object and the background. Both methods do not work well with films, since the chromakey in them is done with much postprocessing and it could not be identified in the absence of artifacts.

5 CONCLUSION

Video classification is becoming increasingly relevant in the modern world due to the large amount of video content and its various categories. In this paper, we described a method to analyze a video sequence using machine learning algorithms and a new approach to data preprocessing. This approach has been successfully implemented and tested on the searching and classifying chromakey in video problem. The results outperformed existing algorithms. Our approach can be easily applied to other video classification problems and can also be scaled to estimate larger video spans depending on the specific task. In the future, it is planned to conduct more qualitative comparisons of different length chromakey sequences. There are many ways to use chromakey technology and each of them has its own specifistic and artifacts. In the future, it is planned to study in detail all chromakey types and select the necessary preprocessing for each type individually, which will help to manage edge cases better.

ACKNOWLEDGEMENTS

The research was supported by the ITMO University, project 623097 ”Development of libraries containing perspective machine learning methods”.

REFERENCES

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. (2021). Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846.

Bertasius, G., Wang, H., and Torresani, L. (2021). Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Ding, L. and Goshtasby, A. (2001). On the canny edge detector. *Pattern recognition*, 34(3):721–725.

Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., Han, X., Chen, Y.-W., and Wu, J. (2020). Unet 3+: A full-scale connected unet for medical image segmentation. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1055–1059. IEEE.

Kaur, A. and Kranthi, B. (2012). Comparison between ycbcr color space and cielab color space for skin color segmentation. *International Journal of Applied Information Systems*, 3(4):30–33.

Koonce, B. and Koonce, B. (2021). Efficientnet. *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, pages 109–123.

Li, J. P., Haq, A. U., Din, S. U., Khan, J., Khan, A., and Saboor, A. (2020). Heart disease identification method using machine learning classification in e-healthcare. *IEEE access*, 8:107562–107582.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66.

Singh, G. and Singh, K. (2022). Chroma key foreground forgery detection under various attacks in digital video based on frame edge identification. *Multimedia Tools and Applications*, pages 1–28.

Song, J.-D., Kim, Y.-G., and Park, T.-H. (2019). Smt defect classification by feature extraction region optimization and machine learning. *The International Journal of Advanced Manufacturing Technology*, 101:1303–1313.

Targ, S., Almeida, D., and Lyman, K. (2016). Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*.

Tong, Z., Song, Y., Wang, J., and Wang, L. (2022). Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093.

Xie, Z., Lin, Y., Yao, Z., Zhang, Z., Dai, Q., Cao, Y., and Hu, H. (2021). Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.-H., Tay, F. E., Feng, J., and Yan, S. (2021). Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 558–567.

Zhang, D. and Zhang, D. (2019). Wavelet transform. *Fundamentals of image data mining: Analysis, Features, Classification and Retrieval*, pages 35–44.