ИНФОРМАТИКА

COMPUTER SCIENCE

МРНТИ 28.23.15
УДК 004.93

https://doi.org/10.51889/2021-4.1728-7901.08

COMPARISON OF FACE DETECTION TOOLS

Amirgaliyev Ye.[1], Sadykova A.[2], Kenshimov Ch.[1]*

[1]"Institute of Information and Computing Technologies" Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan, Almaty, Kazakhstan
[2]Al-Farabi Kazakh National University, Almaty, Kazakhstan
**e-mail: sadykova.akmanat@gmail.com*

Abstract

This article examines the assessment of tools such as Dlib, OpenCV, MTCNN, FaceNet for face recognition. In the process of work, the execution time and the count of detecting of each tool were determined and calculated. The results pictured in graph choose the right tool according to the data obtained in the article that was optimal for next research works. The choice was made for the ease of writing a parallel algorithm. The rationale for the choice of the tool is also given according to the parameters of the use of machine resources, which makes it possible to optimally select a machine without additional and large costs. A comparative analysis of each instrument was performed and the results were identified accordingly. Based on the test results, we divided two cases and tried to give recommendations for each of them. The first case is triggered if only quick face detection is considered in the video. The second case is triggered if more faces are viewed in the video. It turned out that in the first case, we need to use the Dlib tool. In the second case, we can choose tools like Facenet or Mtcnn. The results obtained in the process of the research are presented in the form of graphs, tables and recorded in the conclusion section of this article.

Keywords: face recognition, Dlib, OpenCV, MTCNN, FaceNet, face recognition tools.

Аңдатпа

Е. Амиргалиев[1], А. Садыкова[2], Ч. Кеншимов[1]
[1] Қазақстан Республикасы Білім және ғылым министрлігінің Ғылым комитеті «Ақпараттық және есептеу технологиялары институты», Алматы қ., Қазақстан
[2]әл-Фараби атындағы Қазақ ұлттық университеті, Алматы қ., Қазақстан

БЕТТІ АНЫҚТАУ ҚҰРАЛДАРЫНЫҢ ЖҰМЫСЫН САЛЫСТЫРУ

Бұл мақалада Dlib, OpenCV, MTCNN, FaceNet сияқты тұлғаны тануға арналған бағалау құралдары қарастырылған. Жұмыс барысында әр құралдың орындалу уақыты мен анықталу саны анықталды және есептелді. Тест нәтижелері бойынша біз екі жағдайды бөліп қарастырдық және олардың әрқайсысына ұсыныстар беруге тырыстық. Егер бейнеде тек бетті жылдам анықтау қарастырылса, бірінші жағдай іске қосылады. Егер бейнеде көбірек тұлғалар қаралса, екінші жағдай іске қосылады. Нәтижеге сүйенсек, бірінші жағдайда бізге Dlib құралын қолдану қажет болып шықты. Ал екінші жағдайда біз Facenet немесе Mtcnn сияқты құралдарды таңдай аламыз. Әр құралға салыстырмалы талдау жасалып, нәтижелері сәйкесінше анықталды. Осылайша, біз мақалада алынған мәліметтерге сәйкес біз үшін оңтайлы құралды таңдай алдық. Параллельді алгоритмді жазудың қарапайымдылығы үшін де таңдау жасалды. Құралды таңдаудың негіздемесі сонымен қатар машина ресурстарын пайдалану параметрлеріне сәйкес келтірілген, бұл машинаны қосымша және үлкен шығынсыз оңтайлы таңдауға мүмкіндік береді. Зерттеу барысында алынған нәтижелер графиктер, кестелер түрінде ұсынылған және осы мақаланың соңғы бөлімінде жазылған.

Түйін сөздер: бетті тану, Dlib, OpenCV, MTCN, FaceNet, бетті тану құралдары.

Аннотация

Е. Амиргалиев[1]*, А. Садыкова*[2]*, Ч. Кеншимов*[1]
[1]*Комитет науки Министерства образования и науки Республики Казахстан «Институт информационных и вычислительных технологий», г. Алматы, Казахстан*
[2]*Казахский национальный университет имени аль-Фараби, г. Алматы, Казахстан*

СРАВНЕНИЕ РАБОТЫ ИНСТРУМЕНТОВ ОПРЕДЕЛЕНИЯ ЛИЦА

В этой статье рассматривается оценка таких инструментов, как Dlib, OpenCV, MTCNN, FaceNet для распознавания лиц. В процессе работы определялись и рассчитывались время выполнения и количество обнаружений каждого инструмента. Был проведен сравнительный анализ каждого инструмента, и результаты были соответственно определены. По результатам тестирования мы разделили два случая и попытались дать рекомендации по каждому из них. Первый случай вызывается, если в видео рассматривается только быстрое обнаружение лиц. Второй случай вызывается, если в видео рассматривается больше лиц. Оказалось, что первом случае нам нужно использовать инструмент Dlib. Во втором случае мы можем выбрать такие инструменты, как Facenet или Mtcnn. Таким образом, мы смогли выбрать нужный инструмент по оптимальным для нас, полученным данным в статье. Выбор также производился по лёгкостью написания параллельного алгоритма. Результаты, полученные в процессе исследования, представлены в виде графиков, таблиц и зафиксированы в заключительном разделе данной статьи.

Ключевые слова: распознавания лиц, Dlib, OpenCV, MTCNN, FaceNet, инструменты распознавания лиц.

Introduction

Everyone knows that today face detection is becoming more and more important in our society. It is used in a variety of security applications as well as in a variety areas like marketing, healthcare, etc.

A face recognition system [1] is a technology that allows you to match a human face in a digital image or video frame with a database of faces. Researchers are currently developing several methods of how facial recognition systems work. The most advanced facial recognition method, which is also used to authenticate users with identity verification services, works by accurately identifying and measuring facial features from a given image.

Initially it was a form of a computer application, but more recently, face recognition systems have gained more widespread use on smartphones and in other forms of technology such as robotics. Since computerized face recognition involves measuring the physiological characteristics of a person, face recognition systems are classified as biometric. Although the accuracy of facial recognition as a biometric technology is lower than iris recognition and fingerprint recognition, it is widely used due to its non-contact and non-invasive process. Facial recognition systems have been deployed for enhanced human-computer interaction, video surveillance and automatic image indexing.

In recognition method case, people can recognize faces without much effort [2], face recognition is a complex problem of pattern recognition in computing. Facial recognition systems attempt to identify a human face that is three-dimensional and changes appearance based on lighting and facial expressions based on its two-dimensional image. To accomplish this computational task, face recognition systems perform four stages. First person detection is used to separate the face from the background of the image. In the second step, the segmented face image is aligned based on the face's pose, image size and photographic properties such as lighting and grayscale. The purpose of the alignment process is to accurately localize facial features in the third step, facial extraction. Details such as eyes, nose and mouth are precisely defined and measured in the image to represent the face. The vector of facial features established in this way is then compared at the fourth stage with the database of faces [3].

There are various tools on the market for video processing in the context of image recognition [4]. It is not always clear how different algorithms perform compared to each other or how they cope. This article will compare several of the above mentioned in the abstract section of the tools for determining the human face in two experiments. The first experiment will be a quantitative study that will run different tools with different face reference algorithms. The second experiment focuses on the execution time and effectiveness of a particular tool. As a result of two experiments, it will be possible to answer the questions of which tool and when to use.

Tools for face detecting for today

Within the framework of this article, we consider such face recognition tools as Dlib, OpenCV, MTCNN, FaceNet.

First one is Dlib [5] and it is a modern C ++ toolkit containing machine learning algorithms and tools for building complex C ++ software to solve real-world problems. It is used in both industry and academia in a wide range of fields including robotics, embedded devices, mobile phones, and large high-performance computing environments. Dlib's open source licensing allows it to be used in any application for free [6].

Second one is OpenCV [7] and it is an open source library of computer vision and machine learning software. OpenCV was created to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. OpenCV is a BSD-licensed product that makes code easy to use and modify by enterprises. The library contains over 2500 optimized algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in a video, track camera movements, etc. The library is widely used by companies, research groups and government agencies [8].

Third one is MTCNN [9] or Multi-Task Cascaded Convolutional Neural Networks is a neural network which detects faces and facial landmarks on images. It was published in 2016 by Zhang et al [10]. It is one of the most popular and most accurate face detection tools today. It consists of 3 neural networks connected in a cascade. This method based on deep convolution neural network [11] that is to say, this method can accomplish the task of face detection and alignment at the same time. Compared with the traditional method [12], MTCNN has better performance, can accurately locate the face, and the speed is also faster, in addition, MTCNN can also detect in real time[13].

Last one is FaceNet [14] and it is a deep neural network used to extract details from an image of a person's face. It was published in 2015 by Google researchers Schroff et al [15]. This tool takes an image of a person's face as input and outputs a vector of 128 numbers that represent the most important facial features. In machine learning, this vector is called embedding, because all the important information from the image is embedded in this vector. Essentially, FaceNet takes a person's face and compresses it into a vector of 128 numbers. Ideally, nesting of the same face is also similar.

There are various studies comparing the aforementioned first two instruments in different categories and with other well-known instruments like keras and tensorflow. Of the many articles that were read during the research, the following can be said. The cases can be divided into two. In the first general case, in most applications we do not know in advance the size of the face in the image. Thus, it is better to use the OpenCV method as it is quite fast and very accurate even for small faces. It also detects faces from different angles. It is recommended to use OpenCV in most cases. And in the second case, for medium and large images, Dlib is the fastest method on the CPU. But it does not recognize small faces. So, if you know that your app won't work with very small faces, like a selfie app, then a long-range face detector is the best option. Also, if you can use the GPU, then Dlib face detector is the best option as it is very fast on the GPU and also provides multi-angle detection. But it should be noted that the last two tools in this article are relatively new and require study and comparison.

Each instrument is unique in its own way, but there are also similarities. For example, similarities in the method of application. And also all tools support Python.

If write about the preparatory processes, then you should mention the installation of libraries, the definition and configuration of the input data. As mentioned above in the introduction, this article deals with two experiments. In both cases, a video with a duration of 1 min 10 sec was used for testing. In the first experiment, we run the code and count the number of faces found in the video, as well as the number of coincidences of finding this tool with others. The results of the experiment were recorded in a table and described in a graph. In the second experiment, the code is also run and this time we count the execution time of the code of each tool. The results of the experiment were also recorded in a table and described in a graph. The results of the experiment can be found in more detail below in the results section.

Results after testing

This part shows the results of the testing of each detecting tools. The results are depicted by the tables and graphs.

All 5 tools tested in one video. For each instrument, all detections are combined. It is important to note that this is a quick analysis to resolve the role of the thumb. The result is definitely worth it.

The result of each tool gave us two points, the two points we used to draw a rectangle around each face. Then we examined these points and checked if there is an overlap of these rectangles between the tools.

The following table 1 illustrates our results. It shows how many detections of one instrument overlap with detections of other instruments. In addition, along the main diagonal, we also calculated the total number of detections.

In the process of research for the first experiment, the code was run and tested for comparative analysis. The following results for the number of face detection were obtained and recorded in table 1.

Table 1. The amount of face detection

	Dlib	*OpenCV Haarcascade*	*OpenCV DNN*	*MTCNN*	*FaceNet*
Dlib	*1766*	*477*	*498*	*822*	*714*
OpenCV Haarcascade	*477*	*1504*	*486*	*1021*	*798*
OpenCV DNN	*498*	*486*	*1670*	*736*	*709*
MTCNN	*822*	*1021*	*736*	*1779*	*1195*
FaceNet	*714*	*798*	*709*	*1195*	*1793*

This table 1 shows that as a result, all tools made approximately the same number of detections. Facenet and Mtcnn definitely have the most matches, 1195 identical matches, while Dlib and OpenCV Haar only have 477 matches. Figure 1 clearly visualizes what is presented in Table 1. This graph shows the ratio of amount of each tools face detection case.

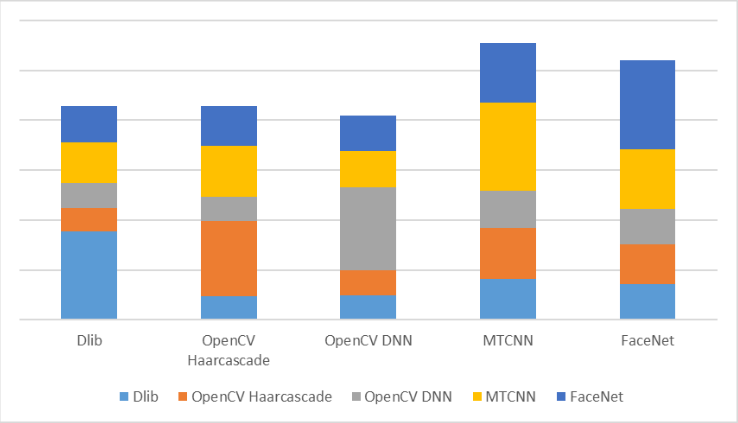

Figure 1. The ratio of amount of face detection

One of the most important characteristics that is important to us is the speed of the tool. Therefore, when we want to choose between face detection tools, depending on our application, runtime can be critical to us. In the following table 2 and figure 2, we have compared the total time it takes for the tools to process the video.

Table 2. Execution time of tools

Dlib	*OpenCV Haarcascade*	*OpenCV DNN*	*MTCNN*	*FaceNet*
122	*351*	*196*	*164*	*191*

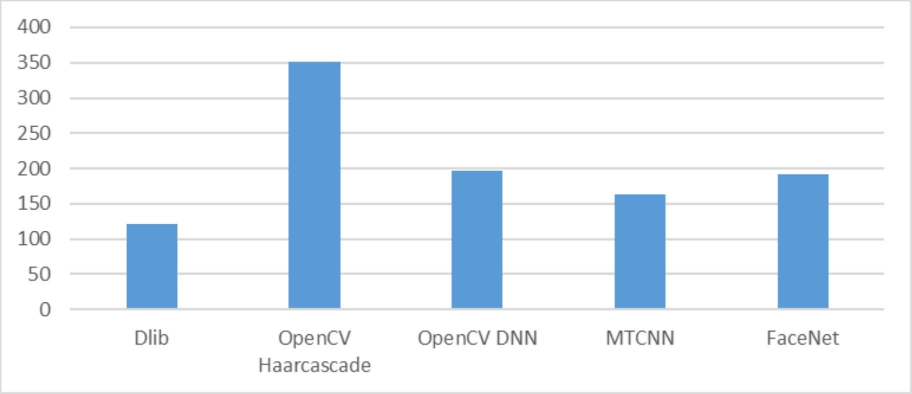

Figure 2. Execution time of tools

The result of testing of each tool clearly shows that the Dlib tool took the shortest time to process the video. On the opposite side, the Harcascade OpenCV tool took the longest time to process the same video. In the next part of this article, the results were concluded overall.

Conclusion

Nowadays there are so many tools in the sphere of face detecting. The idea of testing of these tools appeared to define a faster and more reliable tool. This tool needed to finding by the reason of to use in feature ideas for creating a faster tool for video processing by using parallel methods than the existing on the market today. By these day require in the case of video processing the speed is one of important tool characteristics. The main reason can be the format of video. This format means that the work and processing a big data, by the reason that each video will divide into the many frames in the process.

During the experiment, we analyzed various face recognition algorithms for the same video. The tables above show how many detections each tool made, in addition to their execution time. Based on the test results, we divided two cases and tried to give recommendations for each of them. The first case is called if only fast face detection in the video is considered. The second case is called if more faces in the video are considered. It turned out that for these two cases, which are described above, different tools are used. For example, for the first case, we have to use the Dlib tool. In the second case, we can choose tools such as Facenet or Mtcnn. But if we draw some conclusion to the work done, then still today the choice of using Dlib as a video processing tool will be effective. It should be noted that the word “effective” refers to the speed and the optimal number of face detecting when processing video.

This research is part of a large study and search for effective algorithms for recognition systems. Future plans include using the obtained test results to create a more efficient tool using parallel methods and algorithms.

Acknowledgment

This research has been funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (Grant No. GF AP0805034).

References:

1 S. Zhao, L. Xu, J. Zhang, S. Zhang, X. Chen and J. Wei, "Combine Assembly Quality Detection Based on Multi-Entropy Data Fusion and Optimized LSSVM," in IEEE Access, vol. 9, pp. 63188-63198, 2021, doi: 10.1109/ACCESS.2021.3074255.

2 N. Kohli, D. Yadav and A. Noore, "Face Verification with Disguise Variations via Deep Disguise Recognizer," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018, pp. 17-177, doi: 10.1109/CVPRW.2018.00010.

3 J. Liu, Y. Feng and H. Wang, "Facial Expression Recognition Using Pose-Guided Face Alignment and Discriminative Features Based on Deep Learning," in IEEE Access, vol. 9, pp. 69267-69277, 2021, doi: 10.1109/ACCESS.2021.3078258.

4 H. Sun, Y. Yu, K. Sha and B. Lou, "mVideo: Edge Computing Based Mobile Video Processing Systems," in IEEE Access, vol. 8, pp. 11615-11623, 2020, doi: 10.1109/ACCESS.2019.2963159.

5 H. Xia and C. Li, "Face Recognition and Application of Film and Television Actors Based on Dlib," 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2019, pp. 1-6, doi: 10.1109/CISP-BMEI48845.2019.8965869.

6 I. Lashkov, A. Kashevnik, N. Shilov, V. Parfenov and A. Shabaev, "Driver Dangerous State Detection Based on OpenCV & Dlib Libraries Using Mobile Video Processing," 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), 2019, pp. 74-79, doi: 10.1109/CSE/EUC.2019.00024.

7 L. Xu, W. Shang, W. Lin and W. Huang, "License Plate Detection Methods Based on OpenCV," 2021 21st ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Winter), 2021, pp. 11-16, doi: 10.1109/SNPDWinter52325.2021.00012.

8 E. Cervera, "GPU-Accelerated Vision for Robots: Improving System Throughput Using OpenCV and CUDA," in IEEE Robotics & Automation Magazine, vol. 27, no. 2, pp. 151-158, June 2020, doi: 10.1109/MRA.2020.2977601.

9 D. Gyawali, P. Pokharel, A. Chauhan and S. C. Shakya, "Age Range Estimation Using MTCNN and VGG-Face Model," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225443.

10 Y. Zhang, C. Li, J. Li, G. Wu and H. Huang, "Discrete Model Solving Time-Dependent Matrix Eigen Problem with ZeaD (Zhang et al Discretization) Formula Using 7 Points," 2018 37th Chinese Control Conference (CCC), 2018, pp. 8628-8633, doi: 10.23919/ChiCC.2018.8483264.

11 J. Zhao, H. Guo, Y. Liu, N. Dong and X. Wang, "Tai Le Character Recognition Using Deep Convolution Neural Networks," 2021 International Conference on Computer Communication and Artificial Intelligence (CCAI), 2021, pp. 6-9, doi: 10.1109/CCAI50917.2021.9447467.

12 I. O. Ertugrul, L. A. Jeni, W. Ding and J. F. Cohn, "AFAR: A Deep Learning Based Tool for Automated Facial Affect Recognition," 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), 2019, pp. 1-1, doi: 10.1109/FG.2019.8756623.

13 N. Zhang, J. Luo and W. Gao, "Research on Face Detection Technology Based on MTCNN," 2020 International Conference on Computer Network, Electronic and Automation (ICCNEA), 2020, pp. 154-158, doi: 10.1109/ICCNEA50255.2020.00040.

14 W. Wan and H. J. Lee, "FaceNet Based Face Sketch Recognition," 2017 International Conference on Computational Science and Computational Intelligence (CSCI), 2017, pp. 432-436, doi: 10.1109/CSCI.2017.73.

15 S. Ahmad and B. Fuller, "ThirdEye: Triplet Based Iris Recognition without Normalization," 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS), 2019, pp. 1-9, doi: 10.1109/BTAS46853.2019.9185998.