Вісник Тернопільського національного технічного університету
https://doi.org/10.33108/visnyk_tntu
Scientific Journal of the Ternopil National Technical University
2022, № 3 (107) https://doi.org/10.33108/visnyk_tntu2022.03
*ISSN 2522-4433. Web: **visnyk.tntu.edu.ua***

UDC 378.14

AUTOMATED AI-BASED PROCTORING FOR ONLINE TESTING IN E-LEARNING SYSTEM

Oleh Shkodzinsky; Mykhailo Lutskiv

Ternopil Ivan Puluj National Technical University, Ternopil, Ukraine

Summary. *Based on the analysis of existing on the market algorithmic solutions for identity verification during knowledge control in electronic learning systems, the requirements for the target system were formed. The main algorithms and approaches to the detection and recognition of faces were considered, as a result of which an effective combination of algorithms was chosen. The system of photo fixation and identity verification during knowledge control in LMS ATutor was designed and implemented. Its effectiveness was verified on the basis of a sample of test passes during its work in the real conditions of the educational process. Conclusions were made regarding the feasibility of implementation.*

Key words: *face recognition, photo fixation, knowledge testing, image recognition algorithms, person identification, identification accuracy.*

https://doi.org/10.33108/visnyk_tntu2022.03.076 *Received 21.06.2022*

Statement of the problem. Informatization of education is an important component of the processes of penetrating information technologies into the modern society life. Computer technologies are becoming an integral part of the educational process, which creates prerequisites for its transformation and increased efficiency. Online learning can become extremely effective thanks to the ability to analyze data about students in real time and influence the educational process in the desired direction based on the results of this analysis. Artificial intelligence technologies pave the shortest path from digital representation to the transformation of the learning process itself.

Analysis of recent research results. Along with the positive impact of information technologies on the educational process, new opportunities have appeared due to an unscrupulous attitude to the passing of knowledge test control, especially in the conditions of distance learning, when the persons passing the control are dispersed in space and are outside the visual observation of the examiner [1]. This requires the use of proctoring actions as additional tools and measures for monitoring the course of the testing process, which would provide confirmation of the integrity of behavior for each participant. Despite the active development of such tools over the past few years, the term «proctoring» itself remains not always clearly specified. Nevertheless, a general definition may sound like this: proctoring is an automated procedure for monitoring and controlling remote testing. In recent years, dozens of software products have appeared on the market to solve the mentioned problem, among which the best ones include [2]: «Honorlock», «Talview», «Examity», «Mercer Mettl Online Examination and Proctoring Solutions», etc. These programs are able to analyze a large amount of input data, namely: video, audio and events in the respondent's browser; as well as based on special algorithms and trained models to detect and fix, including in real time, integrity violations. However, despite a number of advantages, each of these programs, which are mainly presented on the market as services, has its own disadvantages. The main disadvantages include: the difficulty or impossibility of integrating with the databases of existing learning management systems (LMS) and their standard testing tools, which have been used by educational institutions for a long time. The need to protect personal data plays an important role here.

The objective of the work is to investigate the known methods of identity verification and to develop an integrated automated system of proctoring for knowledge control, the effectiveness of which would be confirmed in the real conditions of operation of common learning management systems and when using users' standard software and hardware, which would significantly minimize the time and costs of implementation such a system.

Formulation of the problem. Image face detection, which is a variation of the general object detection problem, can be defined as determining whether a given image contains faces, and if it does, finding the location of each face [3]. Face detection is a key task, as it is a necessary step for solving other tasks, such as localization of faces, recognition of faces, analysis of faces, verification of faces, labeling and extraction of faces, tracking of faces, recognition of emotions and facial expressions [4–10].

Nowadays, there are several dozens of computer-based face detection and recognition methods [10]. However, these methods do not provide 100% reliability of identification and, at the same time, often have limitations in recognition performance. Among the main challenges and problems that arise when implementing human face recognition algorithms are:

- the illumination of the recognition object is not always satisfactory;
- variable expressions of emotions on faces;
- different types of skin tones;
- varying the distance to the recognition object;
- variable face orientation;
- complex background;
- the presence of several faces in one image;
- partial covering of faces with glasses, elements of clothing, hands, hair, medical masks, etc.;
- insufficient resolution of camera, etc.

At the first stage of recognition, the face in the image is detected and localized (finding the coordinates). The best results are achieved when the person is looking directly into the video or photo camera while shooting, but modern algorithms also allow face detection in situations where the person is not looking directly into the camera. The result of detection and localization is the found coordinates of the face(s) and its dimensions.

At the next stage, in which the basic parameters are determined, the face image is aligned and normalized geometrically and in terms of brightness, the face image is encoded into a set of basic parameters with the formation of a parametric vector (array). After that, direct recognition takes place – a comparison of the calculated parametric vectors with vectors of already identified persons located in the database.

The most common facial recognition algorithms available today are:

- elastic graph matching;
- principal component analysis, PCA;
- Viola-Jones algorithm;
- HOG (Histogram of Oriented Gradients) algorithm and its combination with SVM (Support Vector Machine) classifier;
- Deep Convolutional Neural Networks, Deep CNNs: AlexNet, VGG, ResNet etc.

The requirements for the algorithms for the proctoring system project are:

- to ensure accuracy of identification – no less than 95%;
- recognition of one frame from the web camera should last no more than 1 second.

The main difference between all the algorithms mentioned above is the face detection mechanism and the method of calculating the basic parameters, that is, translating the face image into a parametric vector.

Taking into account all of the above, to improve the quality of the system being developed, it may be relevant to create hybrid methods that would combine the advantages of several considered algorithms. Thus, combining the faster method of the Histograms of Oriented Gradients (HOG) algorithm in pair with the Support Vector Machines (SVM) and a slower but more accurate algorithm based on deep convolutional neural networks (Deep CNNs) for cases where the first one does not work satisfactorily will allow to create a more effective detector.

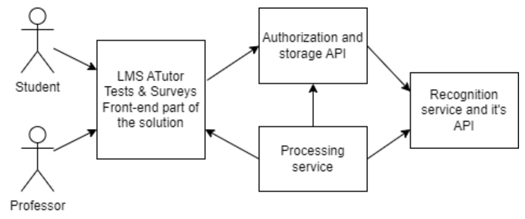
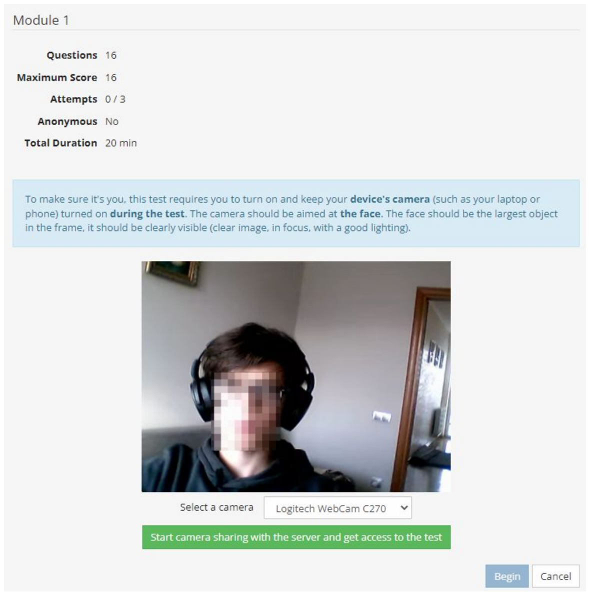

Student
Professor
LMS ATutor Tests & Surveys Front-end part of the solution
Authorization and storage API
Processing service
Recognition service and it's API

Figure 1. Solution architecture diagram


Module 1

Questions	16
Maximum Score	16
Attempts	0 / 3
Anonymous	No
Total Duration	20 min

To make sure it's you, this test requires you to turn on and keep your **device's camera** (such as your laptop or phone) turned on **during the test**. The camera should be aimed at **the face**. The face should be the largest object in the frame, it should be clearly visible (clear image, in focus, with a good lighting).

Select a camera: Logitech WebCam C270

Start camera sharing with the server and get access to the test
Begin
Cancel

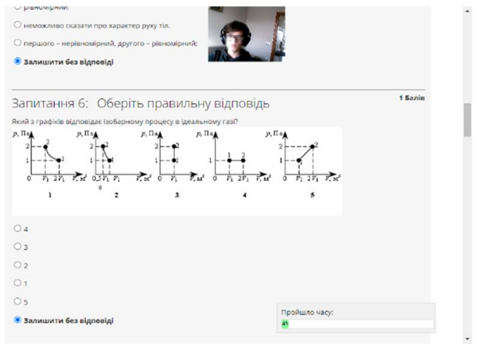
Figure 2. The window for selecting a camera device and granting broadcast permission

Taking into account the fact that it is the type of Deep CNN - residual neural networks (ResNet) that have overcome the human level of image classification, and also dominated in ImageNet competitions for several years [15], and now demonstrate high recognition accuracy at sufficient speed – the recognition stage in this solution is expedient to be implemented using ResNet.

Development results. The proposed solution for photo fixation and automatic identity verification is technically implemented in the form of a group of services that interact with each other using the REST API and integration with the ATutor learning management system (Fig. 1).

The front-end part of the solution is closely integrated with the «Tests and Surveys» module of LMS ATutor and is implemented in JavaScript, PHP, using the jQuery library, Twitter Bootstrap. Interaction with the respondent's camera is implemented using Media Capture and Streams API, which is supported by most modern browsers.

Before starting the test, the student needs to select and grant access to the camera device and agree to send frames from the camera to the server (Fig. 2). Frames are authorized using the authorization API, which issues an access token to the test, after receiving it (and only then) the respondent can start taking the test.



рівномірний;

○ неможливо сказати про характер руху тіл.

○ першого – нерівномірний, другого – рівномірний;

[✓ Залишити без відповіді]

Запитання 6: Оберіть правильну відповідь

1 Балів

Який з графіків відповідає ізобарному процесу в ідеальному газі?

○ 4

○ 3

○ 2

○ 1

○ 5

[✓ Залишити без відповіді]

Пройшло часу:

45

Figure 3. Test run window with camera frames widget

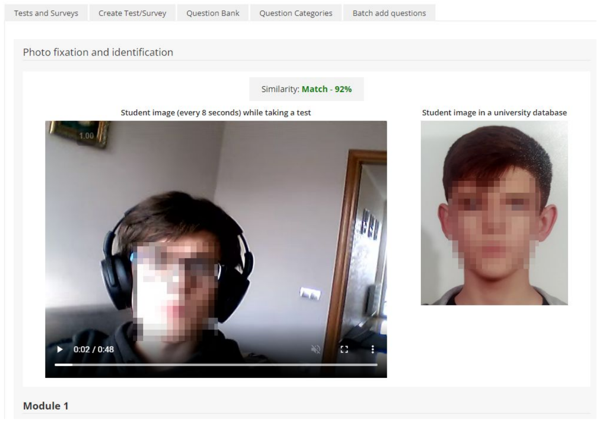


Figure 4. Conclusion on the degree of similarity

During the test, the student sees his image (from the camera frame) in a widget fixed at the top of the window with the test material (Fig. 3) and has the opportunity to adjust the location of the video camera or lighting if the face is out of the frame or other changes have occurred that would make recognition impossible.

Table 1

Main resources and methods of the storage and authorization API

Denotation	Type	Description
	POST/auth	
Parameters		
test_id	int	Test ID
camera_frame	data:image/jpeg	The image from the camera encoded in the data:URI (jpeg) scheme according to the RFC2397 standard
member_id	int	User ID
Result		

200	json	Error code and access token:
		{
		“err_code”: 0,
		“access_token”: “...”
		}

	POST/frame	
Parameters		
access_token	string	Access token
test_id	int	Test ID
camera_frame	data:image/jpeg	The image from the camera encoded in the data:URI (jpeg) scheme according to the RFC2397 standard
member_id	int	User ID
Результат		

200	json	Error code and result message:
		{
		“err_code”: 0,
		“msg”: “...”
		}

During the course of taking a test, frames are sent to the server with a certain frequency using the authorization and storage API, which writes the images to disk in jpeg format for further processing.

The authorization and storage API has two main resources: /auth and /frame. Table 1 describes their methods, parameters and results. The /auth resource authorizes the respondent to pass the test control using a camera frame. The /frame resource receives and saves frames from the camera to storage on the server.

After passing the test, the processing service compresses saved frames from the camera with the VP9 codec, preserving high quality, and packs them into a WebM container. This allows to reduce disk costs, compared to storing in jpeg files, by 7-9 times. The VP9 codec and the WebM container are chosen due to a high level of compression with a high level of quality preservation, a better size/quality ratio than competitors (H.264/MP4, HEVC/MP4), and a completely open license. Packaging and compression are implemented using the FFmpeg suite of programs and libraries. The used ffmpeg command:

```
ffmpeg -framerate 1 -i img%05d.jpg -c:v libvpx-vp9 -r 30 -pix_fmt yuv420p [the path to the container] > /dev/null 2>&1.
```

Table 2

Main resources and methods of the recognition API

Denotation	Type	Description
POST /api/verify_frames		
Parameters		
api_key	string	Access key
source_image1 source_image2 ... source_imageN	image/jpeg	Image frames from the camera in jpeg format
target_image	image/jpeg	A photo of the student from the university's database to be matched with
Tolerance	float	Sensitivity
Result		

200	json	Error code, message, recognition result, best match, and Euclidean Distances for all frames:
		{
		"err_code": 0,
		"msg": "...",
		"verify_result": true/false,
		"best_face_distance": "source_imageN",
		"face_distances": {"source_image1": float, "source_image2": float, ...}
		}

POST /api/face_location		
Parameters		
api_key	string	Access key
source_image	image/jpeg	Image from the camera in jpeg format
Result		

200	json	Error code, message, array of rectangle coordinates with found face(s):
		{
		"err_code": 0,
		"msg": "…",
		"face_location": [[x,y,w,h],..]
		}

After packaging, frames will be accessed by extracting them from the container. This stage is also implemented using FFmpeg:

```
ffmpeg -i [the path to the container] -vf select='eq(n\,1)+ eq(n\,9)+eq(n\,17)+...' -vsync 0 video_frame%d.jpg >/dev/null 2>&1,
```

where eq(n\,1)+ eq(n\,9)+eq(n\,17)+... - numbers of required frames.

The packing-unpacking process is almost lossless, which has a positive effect on subsequent recognition using machine learning algorithms.

After that, the processing service sends the frames to the recognition service, where recognition and comparison with the photo of the respondent's face in the ACS database of the educational institution takes place.

Both services are multithreaded, which makes it possible to quickly process a large stream of data. Considering the nature of the load is CPU-intensive, the number of threads is determined by the number of processor cores involved.

The recognition service is implemented in Python using dlib, face_recognition, cv2, and Numpy libraries. The service API is implemented using the Flask microframework. The main resources and methods are given in Table 2.

The POST /api/verify_frames method is intended for recognition - searching for the coincidence/similarity of certain faces with a given face and returns the Euclidean Distances between vector arrays describing these faces. The POST /api/face_location method allows to get the coordinates of the faces in the image. The service code runs on the Gunicorn WSGI server in a Docker container built with all dependencies.

The recognition process begins by reading images into a Numpy array and finding the location of faces on each of them. For this, a combination of two models included in dlib is used: first, a more efficient (faster) one based on the algorithm of histograms of oriented gradients (HOG) and a linear SVM classifier [11]; and if it did not give a positive result, a slower but more accurate one based on Max-Margin (MMOD) convolutional neural network (CNN) [12], which is able to find faces well even in conditions of different viewing angles, poor lighting, partial face overlap by extraneous objects.

After that, face recognition takes place using the dlib_face_recognition_resnet_model_v1 model [13], which is a residual neural network (ResNet) with 29 layers. The model was trained by the developers on a set of 3 million faces and provides an accuracy of 99.38% in the Labeled Faces in the Wild test [14]. As a result of the model, a 128-dimensional vector array describing the face is obtained. Having such arrays for the face of the person in the camera frames and the face in the photo from the database, the Euclidean Distance between them is calculated, on the basis of which a conclusion about the similarity of the faces is formed.

After receiving a response from the recognition service, the processing service publishes a conclusion about the similarity in the results of passing the test in LMS ATutor along with a complete record of frames available for visual review and comparison (Fig. 4).

The system can flexibly scale from a single thread to the number of available CPU cores. Under the conditions of work at Ternopil Ivan Puluj National Technical University (TNTU), the system processes test passes in 4 threads, using up to 2Gb of RAM per thread and spends 10-14 seconds to process one test pass session (10 captured image frames) on an Intel® Xeon® E5-2680 v2 processor. The system can run on any x86-64 processor that supports the SSE4 and AVX instruction set.

Analysis of the obtained results. The developed software was installed and tested at Ternopil Ivan Pulyu National Technical University (TNTU) and is currently being used in test mode (October 2022). During it's work, a sufficiently representative sample was formed (at the time of writing this article, 2,854 students took 22,472 test passes), so that the main statistical metrics of the effectiveness could be assessed with sufficient accuracy.

The methodology for determining these metrics consisted of the following steps:

- 400 sessions are randomly selected from the total number of test runs;
- each of them is reviewed manually and the number of false rejections, false acceptances, valid rejections, valid acceptances is determined;
- based on these and other values, the main system performance metrics (such as accuracy, FAR, FRR, etc.) are determined.

The results of the calculation are presented in the Table 3.

Table 3

Calculated quantitative indicators of system performance

Denotation	Description	Value
TA	number of valid acceptances	382
TR	number of valid rejections	2
FA	number of false acceptances	3
FR	the number of false rejections	13
Total		400

False Reject Rate (*FRR*) and False Acceptance Rate (*FAR*) are one of the most important metrics for evaluating the quality and efficiency of recognition systems.

FRR – false rejection rate – the probability that the system will refuse authentication to a valid (true) user.

FRR is calculated for this case as follows:

$$FRR = \frac{FR}{N} \times 100\% = \frac{13}{400} \times 100\% = 3.25\%, \tag{1}$$

where *FR* is the number of false rejections, *N* is the total number of cases.

FAR – false acceptance rate – the probability that the system mistakenly authenticates an incorrect user. *FAR* is calculated for this case as follows:

$$FAR = \frac{FA}{N} \times 100\% = \frac{3}{400} \times 100\% = 0.75\%, \tag{2}$$

where *FA* is the number of false acceptances, *N* is the total number of cases.

Typical values for good quality 2D face recognition systems are 2.5% for *FRR* and 0.1% for *FAR*. In this case, slightly higher values are explained by a significant number of test runs under unsatisfactory lighting conditions.

Another important metric is *Accuracy* – it shows how many of the acceptances and rejections turned out to be really true or, simply put, how many times system was correct overall:

$$Accuracy = \frac{TA + TR}{TA + TR + FA + FR} \times 100\% = \frac{382 + 2}{382 + 2 + 3 + 13} \times 100\% = 96.0\%, \tag{3}$$

where *TA* is the number of valid acceptances, *TR* is the number of valid rejections.

The accuracy of the used ResNet-29 model in the Labeled Faces in the Wild test is 99.38% [13]. The accuracy of the developed system turned out to be somewhat lower, but at this stage of development is quite satisfactory.

Conclusions. The topical issue of identity verification during knowledge control in electronic learning systems is considered. Strengths and weaknesses of existing solutions were studied, as a result of which an optimal set of requirements for such a system was formed.

Modern approaches and algorithms for face detection and recognition were analyzed and, as a result, an effective combined approach was chosen - to use a fast HOG+SVM algorithm and a slower but more accurate CNN for cases where the first one did not give results. This made it possible to implement a fast detector with a low number of errors. An efficient (fast and accurate enough) model based on ResNet-29 was chosen for recognition. The model justified its choice by demonstrating an accuracy of 96.0% for sufficiently difficult conditions (low lighting, medical masks on faces, etc.).

Based on the formed requirements and selected algorithms, a system was designed and implemented for photo fixation and verification of the person during knowledge control in LMS ATutor. The system has been in use for about a year at TNTU.

The effectiveness of the system was investigated according to a number of metrics (based on a sample of test runs during its operation). Somewhat high metrics of levels of false acceptances (0.75%) and false rejections (3.25%) were revealed. One of the main factors affecting these levels was found to be unsatisfactory lighting when obtaining images of a person. The next step to improve these metrics is the formation of requirements for lighting and frame quality for respondents, as well as preliminary control of image quality before admission to testing and starting identification.

The determined accuracy of the system (96.0%) is high enough to allow to use it as a fairly reliable system for proctoring with functions of providing recommendations, based on which the final decision is made by a human.

References

1. Shkodzinsky O. K., Lutskiv M. M., Smolii I.-M. S. Rozvytok zasobiv veryfikatsii osoby ta yii dii pry kontroli znan v umovakh dystantsiinoho navchannia. Zbirnyk tez dopovidei X Mizhnarodnoi naukovo-praktychnoi konferentsii molodykh uchenykh ta studentiv „Aktualni zadachi suchasnykh tekhnolohii“, 24–25.11.2021. T.: FOP Palianytsia V. A., 2021. Vol I. P. 138–139. (Kompiuterno-informatsiini tekhnolohii ta systemy zviazku). [In Ukrainian].
2. Best Online Proctoring Software, 2020. G2 Bussness Software Reviews. URL: https://www.g2.com/categories/online-proctoring.
3. Yang M. H., Kriegman D. J., and Ahuja N. 2002. Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (1): 34–58. ISSN 01628828. DOI: https://doi.org/10.1109/34.982883
4. Lam, K.-M. and Yan, H., 1994. Fast algorithm for locating head boundaries. Journal of Electronic Imaging, 3 (4): 351–359. ISSN 1017-9909. DOI: https://doi.org/10.1117/12.183806
5. Chi L., Zhang H. and Chen M., 2017. End-To-End Face Detection and Recognition. arXiv preprint, 1703.10818:1-9.
6. Moghaddam B. and Pentland A. 1997. Probabilistic visual learning for object representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19 (7): 696–710. ISSN 01628828. DOI: https://doi.org/10.1109/34.598227
7. Ou W., You X., Tao D., Zhang P., Tang Y., and Zhu Z., 2014. Robust face recognition via occlusion dictionary learning. Pattern Recognition. 47 (4): 1559–1572. ISSN 00313203. DOI: https://doi.org/10.1016/j.patcog.2013.10.017
8. Gao Y. and Qi Y., 2005. Robust visual similarity retrieval in single model face databases. Pattern Recognition, 38 (7): 1009–1020. ISSN 00313203. DOI: https://doi.org/10.1016/j.patcog.2004.12.006
9. Essa I. and Pentland A., 2002. Facial expression recognition using a dynamic model and motion energy. In Proceedings of 5th IEEE International Conference on Computer Vision. 20–23 June 1995.P. 360–367. IEEE. Doi: 10.1109/ iccv.1995.466916.
10. Agrawal S. and Khatri P. Facial Expression Detection Techniques: Based on Viola and Jones Algorithm and Principal Component Analysis, 2015 Fifth International Conference on Advanced Computing & Communication Technologies. 2015. P. 108–112. Doi: 10.1109/ACCT.2015.32.
11. dlib default face detector – dlib documentation. URL: http://dlib.net/python/#dlib.get_frontal_face_detector.
12. King Davis E. 2015 Max-margin object detection. arXiv preprint arXiv:1502.00046.

13. GitHub – davisking/dlib-models: Trained model files for dlib example programs. URL: https://github.com/davisking/dlib-models#dlib_face_recognition_resnet_model_v1datbz2.
14. LFW Face Database. URL: http://vis-www.cs.umass.edu/lfw/.
15. ImageNet Winning CNN Architectures (ILSVRC). URL: https://www.kaggle.com/getting-started/149448. DOI: https://doi.org/10.1109/ACCT.2015.32

Список використаних джерел

1. Шкодзінський О. К., Луцків М. М., Смолій І. М. І. Розвиток засобів верифікації особи та її дій при контролі знань в умовах дистанційного навчання. Актуальні задачі сучасних технологій: зб. тез доп. Х міжнар. наук.-практ. конф. (м. Тернопіль, 24–25 лис. 2021 р.). Тернопіль, 2021. Том I. С. 138–139.
2. Best Online Proctoring Software, 2020. G2 Bussness Software Reviews. URL: https://www.g2.com/categories/online-proctoring.
3. Yang M. H., Kriegman D. J., and Ahuja N., 2002. Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (1): 34–58. ISSN 01628828. DOI: https://doi.org/10.1109/34.982883
4. Lam K.-M. and Yan H., 1994. Fast algorithm for locating head boundaries. Journal of Electronic Imaging, 3 (4): 351–359. ISSN 1017-9909. DOI: https://doi.org/10.1117/12.183806
5. Chi L., Zhang H., and Chen M., 2017. End-To-End Face Detection and Recognition. arXiv preprint, 1703.10818:1-9.
6. Moghaddam B. and Pentland A., 1997. Probabilistic visual learning for object representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19 (7): 696–710. ISSN 01628828. DOI: https://doi.org/10.1109/34.598227
7. Ou W., You X., Tao D., Zhang P., Tang Y., and Zhu Z., 2014. Robust face recognition via occlusion dictionary learning. Pattern Recognition, 47 (4): 1559–1572. ISSN 00313203. DOI: https://doi.org/10.1016/j.patcog.2013.10.017
8. Gao Y. and Qi Y., 2005. Robust visual similarity retrieval in single model face databases. Pattern Recognition, 38 (7): 1009-1020. ISSN 00313203. DOI: https://doi.org/10.1016/j.patcog.2004.12.006
9. Essa I. and Pentland A., 2002. Facial expression recognition using a dynamic model and motion energy. In Proceedings of 5th IEEE International Conference on Computer Vision. P. 360–367. IEEE. Doi: 10.1109/iccv.1995.466916. 20-23 June 1995.
10. S. Agrawal and P. Khatri, Facial Expression Detection Techniques: Based on Viola and Jones Algorithm and Principal Component Analysis, 2015 Fifth International Conference on Advanced Computing & Communication Technologies. 2015. P. 108–112. Doi: 10.1109/ACCT.2015.32.
11. dlib default face detector – dlib documentation. URL: http://dlib.net/python/#dlib.get_frontal_face_detector.
12. King Davis E., 2015 Max-margin object detection. arXiv preprint arXiv:1502.00046.
13. GitHub – davisking/dlib-models: Trained model files for dlib example programs. URL: https://github.com/davisking/dlib-models#dlib_face_recognition_resnet_model_v1datbz2.
14. LFW Face Database. URL: http://vis-www.cs.umass.edu/lfw/.
15. ImageNet Winning CNN Architectures (ILSVRC). URL: https://www.kaggle.com/getting-started/149448. DOI: https://doi.org/10.1109/ACCT.2015.32

УДК 378.14

АВТОМАТИЗОВАНИЙ ПРОКТОРИНГ НА ОСНОВІ ШІ ДЛЯ ОНЛАЙН-ТЕСТУВАННЯ В СИСТЕМІ ЕЛЕКТРОННОГО НАВЧАННЯ

Олег Шкодзінський; Михайло Луцків

Тернопільський національний технічний університет імені Івана Пулюя, Тернопіль, Україна

Резюме. *Методи контролю знань у формі тестування зарекомендували себе як один із перспективних засобів підвищення ефективності управління якістю навчального процесу. З розвитком інформаційних технологій використання тестів у цій галузі вийшло на новий якісний та кількісний рівень. Однак поряд із позитивним впливом інформаційних технологій на* *розвиток інструментів для контролю знань, з’явилися нові можливості для недобросовісного ставлення до проходження тестового контролю, особливо в умовах дистанційного навчання, коли особи, які проходять контроль, розосереджені в просторі та перебувають поза візуальним спостереженням екзаменатора. Це потребує додаткового запровадження прокторингових дій як інструментів і заходів моніторингу перебігу процесу тестування, щоб забезпечували підтвердження чесності поведінки кожного учасника. Переважна більшість технічних вирішень, що стосуються вказаної проблеми, спираються на використання мультимедійних засобів для розпізнавання особи та її дій. Додатковою умовою є те, що такі технічні рішення мають легко інтегруватися з уже існуючими системами управління навчанням (LMS), які тривалий час використовуються закладами освіти. На основі аналізу існуючих на ринку алгоритмічних рішень верифікації особистості під час контролю знань в електронних системах навчання сформовано вимоги до цільової системи. Розглянуто основні алгоритми та підходи до виявлення та розпізнавання облич, у результаті чого обрано ефективну комбінацію алгоритмів. Розроблено та впроваджено систему фотофіксації та перевірки особистості під час контролю знань у LMS ATutor. Ефективність роботи системи перевірено на основі вибірки тестових проходжень, сформованої під час її роботи в реальних умовах навчального процесу. Зроблено висновки стосовно доцільності впровадження й подальшого розвитку.*

***Ключові слова**: розпізнавання обличчя, фотофіксація, перевірка знань, алгоритми розпізнавання зображень, ідентифікація особи, точність ідентифікації.*

https://doi.org/10.33108/visnyk_tntu2022.03.076 *Отримано 21.06.2022*