Hardware-software solution for personnel identification for accounting of working hours in a construction organization

O F Danilov¹, I K Nikitin², A M Osipenko¹ and V O Domanskij¹

¹Industrial University of Tyumen, 38 Volodarskogo st., Tyumen, Tyumenskaya oblast, 625000, Russia

²RCCC Smetny Alyans, 21 Solnechny proezd, Tyumen, Tyumen region, 625022, Russia

E-mail: danilov_of@mail.ru

Abstract. The efficiency of the construction process largely depends on the possibility of reducing costs at all stages. The simplest and most understandable way of saving is the use of methods and models of lean production in such elements as labor productivity, time tracking, reduction in material consumption, rational use of equipment. With the participation of the authors of the article, a system for assessing anomalies in production was developed for one of the construction organizations, which allows, in particular, to monitor the beginning and end of a work shift, the time spent on a specific task, an employee's late arrival and early departures from work. One of the important elements of the system is a solution related to personnel identification - recognition of faces and images from the lens of the built-in camera of a stationary hardware device, which was named "Infokiosk". The article describes the methods used to form the solution. The results are presented in the form of an implemented software and hardware solution.

For any organization, the most important resource is the working time of its employees. In the realities of the modern world, taking into account the constant competition, one of the most important factors that increase the company's ability to compete is the problem of monitoring working hours.

Working time monitoring is a rather laborious process. Therefore, to simplify this procedure and timely organize all employee data in organizations of all sectors and sizes, automated time tracking systems are used.

The basic principle of such systems is to register the total number of working hours of each employee, implement and register all delays, vacations, weekends and sick leaves. At any time, the system allows you to receive reports on all of the above criteria.

The time tracking system also stores statistics and generates reports. Depending on the task, the system can generate reports on different parameters: the beginning and end of the working day, types of labor discipline violations, processing, intervals, employees, departments.

Active monitoring of the work of all employees on the construction site requires a high degree of automation. If we increase the number of engineering and technical workers who will monitor the performance of work and respond as quickly as possible to emerging problems, as well as control discipline, then the construction organization may be able to avoid problems with the timing of the work,
but this will cost significant costs for the payment fund labor of engineering and technical workers, whose salary level is often higher than that of the average worker.

Thus, increasing "manual" control is not the best solution. A much more effective way to solve the problem is to automate the control process, which allows you to track the work of all workers on the construction site in real time, while not hiring technical personnel in a one-to-one ratio with the worker.

To implement the declared functionality of "Infokiosk", a system called "FaceNet" was used, which directly correlates face images with a compact Euclidean space, where the distances directly correspond to the degree of similarity of faces. After matching evaluations, the face recognition, verification and clustering task is implemented using standard methods with FaceNet attachments as feature vectors. [1]

The method uses a deep convolutional network trained to directly optimize the embedding of itself. To train the neural network, the developers use triplets with aligned matching or non-matching face patches generated using the triplet analysis method.

The advantage of this approach is that the comparison is more efficient, since the accuracy of face recognition today is achieved in a photograph with a capacity of 128 bytes per face.

The concept of harmonic attachments and harmonic triplet losses is also used, which describe different variants of edge attachments (produced by different networks), compatible with each other, and allow direct comparison with each other.

As part of the implementation of the minimum working model, a unified system was developed for checking the face of the same person and recognizing who this person is.

The network is trained in such a way that the squares of the L2 distances in space directly correspond to the similarity of the faces. The faces of the same person have short distances, while the faces of different people are much larger.

Once this attachment is done, the aforementioned tasks become simpler: face checking simply involves setting thresholds for the distance between the two attachments. Recognition becomes the task of k-NN classification.

Previous deep network based facial recognition approaches use a classification layer trained from a set of face IDs. They then evaluate an intermediate level of bottlenecks to generalize recognition beyond the set of identifiers used in training.

The disadvantages of this approach are its low efficiency. In this approach, it is necessary to make the assumption that the idea of bottlenecks generalizes well to new faces, and when using a layer with bottlenecks, at least a thousand measurements must be taken.

In contrast to these approaches, FaceNet directly trains its compact 128-D nesting solution using a loss function based on LMNN triplets.
Figure 1. Pose and lighting are the main problems of face recognition.

This figure shows the FaceNet output distances between pairs of faces of the same person and another person in different combinations of pose and lighting. Distance 0.0 means the faces are identical, 4.0 corresponds to the opposite spectrum, two different identities.

Triplets consist of two identical face miniatures and one mismatched one, and the assessment consists in moving the positive pair away from the negative using distance. Thumbnails are narrow, cropped areas of the face, with no 2D or 3D alignment other than scaling and translation.

Choosing which triplets to use turns out to be very important to achieve a good result, and FaceNet introduces a new online negative example analysis strategy that ensures the ever-increasing complexity of triplets while learning online.

To illustrate the complex examples that this method can handle, pairs of images from PIE (Figure 2) are shown that were previously considered impracticable for face recognition systems.

Similar to other works that use deep networks, the approach used is a guided method that learns and shapes its solution directly from the pixels of the face. Instead of using engineering functions, a large dataset with labeled faces is used to derive appropriate differences for pose, lighting, and other variation conditions. [2]

The neural network consists of a layer of periodic input and deep CNN with subsequent L2-normalization, which leads to embedding of edges. This is followed by a triplet loss during training. The structure of the model is shown in Figure 2. [1]

Figure 2. Model structure.
The loss triplet (loss triplet) minimizes the distance between the anchor and a positive result, both of which have the same identity, and maximizes the distance between the anchor and the negative result of the other identity (Figure 3). [1]

![Figure 3. The training cycle of the neural network.](image)

The FaceNet system, as we have already noted, uses a deep convolutional network. Given the details of the model and treating it as a black box (Figure 2), the technique is end-to-end training of the entire system. Namely, the system tends to embed \( f(\mathbf{x}) \) from the image \( \mathbf{x} \) into the feature space \( \mathbb{R}^d \), so that the square of the distance between all faces, regardless of the visualization conditions, which are identical to each other, tends to a minimum, while the square of the distance between images of faces of different people growing.

The embedding is represented as \( f(\mathbf{x}) \in \mathbb{R}^d \). Thus, it embeds the image \( \mathbf{x} \) in \( d \)-dimensional Euclidean space. Moreover, this embedding is restricted to exist on a \( d \)-dimensional hypersphere, \( \|f(\mathbf{x})\|_2 = 1 \). This limitation is motivated in the context of the nearest neighborhood classification. This requires that the \( x_a^i \) (anchor) image of a particular person is closer to all other \( x_p^i \) (positive) images of the same person than any \( x_n^i \) (negative) image of other people. This is indicated in figure 3.

In this way,

\[
\|f(x_a^i) - f(x_p^i)\|_2^2 + \alpha < \|f(x_a^i) - f(x_n^i)\|_2^2 \quad (1)
\]

\[
\forall (f(x_a^i), f(x_p^i), f(x_n^i)) \in T \quad (2)
\]

where \( \alpha \) - is the difference between positive and negative pairs; \( T \) – is the set of all possible triplets in the training set with cardinality \( N \).

Generation of all possible triplets will result in many triplets that are easily satisfied (that is, match the constraint in equation (1)). These triplets will not facilitate learning and may lead to slower convergence as they will go through the network anyway. It is very important to choose hard triplets that are active and therefore can improve the model.

To ensure fast convergence, it is important to choose triplets that violate the triplet constraint in formula (1).

To have a meaningful positive distance anchor, it is necessary to ensure that a minimum number of samples of any identity are present in each mini-batch.

The execution of the recognition task, in our opinion, can be best implemented using an open software library for machine learning called TensorFlow. This library was developed by Google for solving problems of building and training a neural network in order to automatically find and classify images. [1]

The principle of working with TensorFlow is based on drawing up graphs of operations, transferring the necessary data and commands to perform calculations. The figure below (Figure 4) on the left shows a graph that contains only one vertex, representing a constant with a value of 1. The middle graph shows the summation operation.

Upon request, TensorFlow will calculate the values of the edges directed into it and add them. The right graph has two vertices with subtraction and squaring operations. When performing a squaring operation, the subtraction action will be performed first.
Figure 4. TensorFlow graphs.

The TensorFlow framework will be used as the backend of the Keras open-source neural network library for deep learning.

At the center of the neural network is the loss function used to determine the error between the actual data and the resulting results of the neural network.

The goal of training a neural network is to minimize the magnitude of this error. The loss function is used to approach this goal.

Using a loss function called TripletLoss, during training, the distance between the anchor and images with a similar appearance is minimized, and between different images is maximized.

The architecture of the neural network is a Siamese network. It learns to differentiate input data, allowing the neural network to understand which images are identical and which are not (Figure 5). [4]

Figure 5. Scheme of the Siamese architecture of a neural network.

Siamese networks are made up of two identical twin neural networks that have identical weights. The neural network uses one of the two received images as input, and then the outputs of the last layers of each network are sent to a function that determines the content of the same identifiers in the images.

Face recognition, like any biometrics, cannot give absolutely accurate results and has several limitations in the process of work. In particular, the database should not exceed 500 persons. In addition, the orientation of the face relative to the camera, the quality of the IP camera and photos stored in the database, the quality of lighting in the required recognition zone, and the size of the face in the frame (allowable size is 100 pixels) are very important.

The development of the hardware and software solution was carried out in the Python 3.7 programming language, where Spyder was used as an integrated development environment.

Python libraries such as tensorflow, keras, scikit-learn, opencv-python, h5py, matplotlib and pypiwin32 were used for development.

The recognition system uses the TripletLoss algorithm. This algorithm minimizes the distance between the anchor and the photographs stored in the database, which, in turn, contain similar appearances, and maximizes the distance between different appearances.
The recognition system uses the TripletLoss algorithm. This algorithm minimizes the distance between the anchor and the photographs stored in the database, which, in turn, contain similar appearances, and maximizes the distance between different appearances.

Initialization of the face recognition operation occurs from the dimension input (3, 96, 96). This means that the picture is transmitted in the form of three RGB channels and a dimension of 96 × 96 pixels.

The next step was to upload all photos for processing by the system (Figure 6).

![Figure 6. Database of persons for personal identification in the "InfoKiosk" system.](image)

One photo of a certain person in good quality is enough for work. Further, to debug the system and minimize identification errors, retraining was performed. The work setup was as follows:
- we carry out the first compilation of our code and determine the minimum distance (Figure 7);
- we set the distance in the program code;
- we get a working identification code.

![Figure 7. First compilation of a personality recognition system.](image)

To create a program on the Windows platform, the Qt Designer solution in the Python programming language was used.
Initially, a welcome window pops up with the functions "Log in to the system" and "Log out". When you press the "Authorization in the system" button, the program compares the faces with the database and identifies the worker (Figure 8).

Figure 8. Authentication of an employee in the "Infookiosk" system

After face recognition, the system displays a window with information about the employee, and the worker checks his data.

Identity confirmation is the basis for obtaining a tracker and then linking the tracker to the system with fixing the start time of the working day.

The developed solution is an important element of the system for monitoring and managing the efficiency of construction projects based on the analysis of the use of working time and compliance with technological processes.

References
[1] Schroff F Kalenichenko D and Philbin J 2015 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 815-823
[2] Liu X T Chen and J Rittscher 2006 Optimal Pose for Face Recognition *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2 pp 1439–46
[3] Jain A K 1999 *Biometrics: Personal Identification in Networked Society* p 411 doi:10.1007/b117227
[4] Adini Y Moses Y and Ullman S 1997 Face Recognition: The Problem of Compensating for Changes in Illumination *IEEE Transaction on Pattern Analysis and Machine Intelligence* 19(7) pp 721-732
[5] Liu C Wechsler H 2002 Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition *IEEE Transactions On Image Processing* 11(4) pp 467-476
[6] Jenkins R Burton A M 2008 Response to Comment on “100% Accuracy in Automatic Face Recognition” *Science* 321(5891) p 912
[7] Parkhi O M Vedaldi A and A Zisserman 2015 Deep Face Recognitio *British Machine Vision Association*
[8] Shen Y, M Yang, B Wei, C Chou and W Hu 2017 Learn to Recognise: Exploring Priors of Sparse Face Recognition on Smartphones *IEEE Transactions on Mobile Computing* 16(6) pp1705–17
[9] Sui J Y Zhu and Han S 2006 Self-Face Recognition in Attended and Unattended Conditions: an Event-Related Brain Potential Study *Neuroreport* 17 (4) pp 423–27
[10] Tropchenko A A 2015 *Metody vtorichnoi obrabotki i raspoznavania izobrazhenii* p 215
[11] Liu CH, Bhuiyan MA, Ward J and Sui J 2009 Transfer between Pose and Illumination Training
in Face Recognition *Journal of Experimental Psychology. Human Perception and Performance* 35 (4) pp 939–47

[12] Zhong Y R Arandjelovic and Zisserman A 2019 GhostVLAD for Set-Based Face Recognition *Lecture Notes in Computer Science* 11363 pp 35–50

[13] Xie W and Zisserman A P 2018 Multicolumn Networks for Face Recognition *Machine Vision Association*

[14] Patane A and Kwiatkowska M Z 2018 Calibrating the Classifier: Siamese Neural Network Architecture for End-to-End Arousal Recognition from ECG *Lecture Notes in Computer Science* 11331 pp 1–13

[15] Vedaldi A, Novotny D and Larlus D 2018 Capturing the Geometry of Object Categories from Video Supervision *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42(2) pp 261–75