DETECTION OF A HUMAN HEAD ON A LOW-QUALITY IMAGE AND ITS SOFTWARE IMPLEMENTATION

D. Yudin [1, *], A. Ivanov [1], M. Shchendrygin[1]

[1] Dept. of Technical Cybernetics, Belgorod State Technological University named after V.G. Shukhov, Belgorod, Russia - ydin.da@bstu.ru

Commission II, WG II/5

KEY WORDS: Image recognition, Human head, Detection, Deep learning, Convolutional neural network, Software

ABSTRACT:

The paper considers the task solution of detection on two-dimensional images not only face, but head of a human regardless of the turn to the observer. Such task is also complicated by the fact that the image receiving at the input of the recognition algorithm may be noisy or captured in low light conditions. The minimum size of a person’s head in an image to be detected for is 10×10 pixels. In the course of development, a dataset was prepared containing over 1000 labelled images of classrooms at BSTU n.a. V.G. Shukhov. The markup was carried out using a segmentation software tool specially developed by the authors. Three architectures of convolutional neural networks were trained for human head detection task: a fully convolutional neural network (FCN) with clustering, the Faster R-CNN architecture and the Mask R-CNN architecture. The third architecture works more than ten times slower than the first one, but it almost does not give false positives and has the precision and recall of head detection over 90% on both test and training samples. The Faster R-CNN architecture gives worse accuracy than Mask R-CNN, but it gives fewer false positives than FCN with clustering. Based on Mask R-CNN authors have developed software for human head detection on a low-quality image. It is two-level web-service with client and server modules. This software is used to detect and count people in the premises. The developed software works with IP cameras, which ensures its scalability for different practical computer vision applications.

1. INTRODUCTION

Task of people detecting, counting and recognizing is often arising when developing modern video analytics systems for monitoring of housing and business premises, road infrastructure. Its important subtask is to detect the head of a person who may be far away from the camera or turned back to the camera. The most popular methods work effectively only when a person had turned to the camera by face and a head occupies a significant part of the frame. Examples of such approaches are the Viola-Jones method (Viola et al., 2003) or a detector based on histograms of oriented gradients (HOG) (Dalal et al., 2005). Nowadays reliable methods of detecting and recognizing human faces based on deep learning are being widely studied and applied (LeCun et al., 2015).

In this paper we explore various architectures of deep convolutional neural networks for human heads detection.

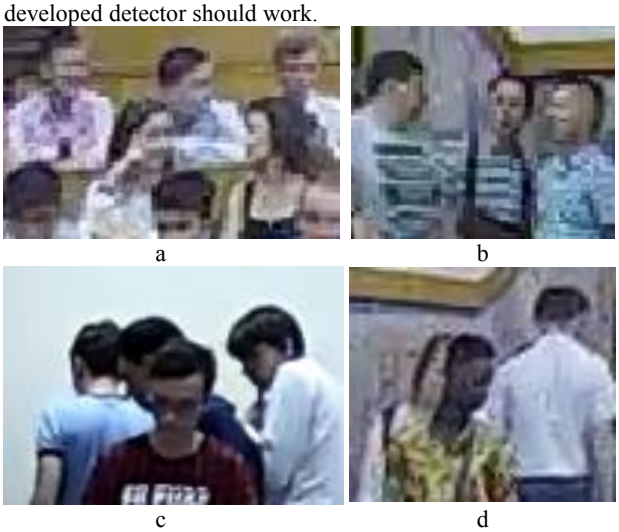
An important area is also the development of software that implements deep learning approaches. For modern applications, it is necessary to analyze and apply the capabilities of popular open-source frameworks, for example, Tensorflow object detection API (Huang et al., 2017) or Mask R-CNN implementation (Waleed, 2017).

However, special attention should be paid to the development and design of application systems with which the end user works. He usually wants to see the results of image recognition and the required statistics in convenient form. In such systems, in addition to the object detection module, much attention is paid to image capturing from one or several cameras, developing a database and creating user interfaces.

2. TASK FORMULATION

This paper considers the detection on two-dimensional images not only face, but head of a person regardless of the turn to the observer. Such task is also complicated by the fact that the image receiving at the input of the recognition algorithm may be noisy or captured in low light conditions. Also, the size of the object (human head) can vary widely in the image. The minimum size of a person’s head in an image to be detected for is 10×10 pixels.

The Fig. 1 shows examples of image fragments with which the developed detector should work.



a b c d

Figure 1. Examples of low-quality images for human head detection task

* Corresponding author

Images of human heads in these images may be very small (Fig. 1a, b), may overlap and apply to people turned their backs to the video camera (Fig. 1c, d).
The main stages of solving the task of human head detection in a low-quality image are:
1) the formation of a suitable dataset;
2) the study of various architectures of convolutional neural networks, allowing to detect the human heads with acceptable quality. For practical applications, it is necessary that the quality measures Precision and Recall (Olson, 2008) exceed the value of 0.9 (with Intersection over Union IoU>0.5). The results of this work are planned to be used primarily for monitoring room attendance. First of all, it is important that the number of false positives be as low as possible (i.e. Precision should be as high as possible). A small number of passes is also desirable, but not so significant, because the system takes the maximum number of people present on the basis of several frames and, if a person was not found on one of the frames, he can be found and counted on others. The mean detection time per frame should not exceed 10 s;
3) software implementation of the system for detecting a human head based on a client-server approach in the form of a web application and testing its performance.

3. DATASET PREPARATION

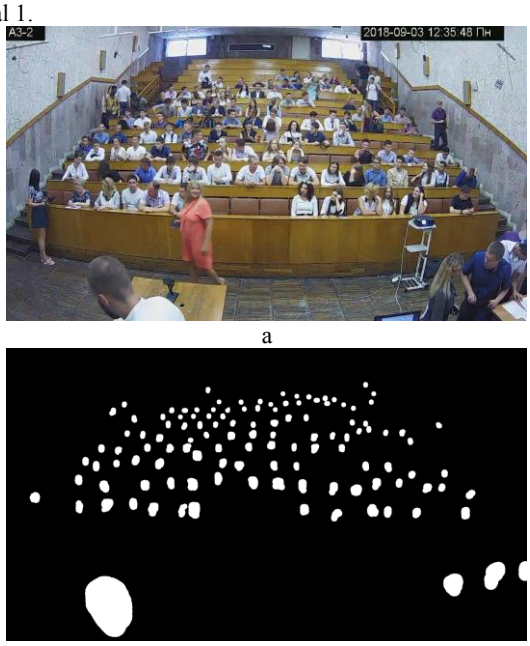
In the course of development, a dataset was prepared containing over 1000 labelled images of classrooms at Belgorod State Technological University named after V.G. Shukhov (BSTU n.a. V.G. Shukhov). Images were taken under various lighting conditions and in the presence of interference and noise. The markup was carried out using a segmentation software tool specially developed by the authors (Yudin, 2018).
The dataset consists of 1280×720 pixel color images (Fig. 2a). The smallest human head size is 10×10, the biggest size is 150×150.
For each of the images a binary mask (reference markup) is assigned (Fig. 2b). If two objects (heads) overlap, then a dividing line with a width of 2 pixels is drawn between them. The number of objects in one image varies from 0 to 140.
Training sample size: 500 images. The test sample also includes 500 images.

4. DEEP NEURAL NETWORKS ARCHITECTURES FOR HUMAN HEADS DETECTION ON A LOW-QUALITY IMAGE

Three architectures of convolutional neural networks were trained for human head detection task: a fully convolutional neural network (FCN) with clustering, similar to that described in (Yudin et al., 2018), the Faster R-CNN architecture (Ren et al., 2015), and the Mask R-CNN architecture (He et al., 2017).

4.1 Fully convolutional neural network (FCN) with clustering

Fig. 3 shows the structure of the detector based on Fully convolutional neural network inspired by (Ronneberger, 2015). The result of the network in the form of a grayscale image is binarized using a manually defined threshold (equal to 100). Then the binarized image is clustered using the fast DBSCAN algorithm (Ester, 1996).
The network training process is described in detail in (Yudin et al., 2018). During it the source color image of 1280×720 pixels and the corresponding binary mask of the same size were fed to the network input and output, respectively. Batch size is equal 1.



a

b

Figure 2. Example of source image and mask from dataset: a – source color image, b – binary mask (markup)

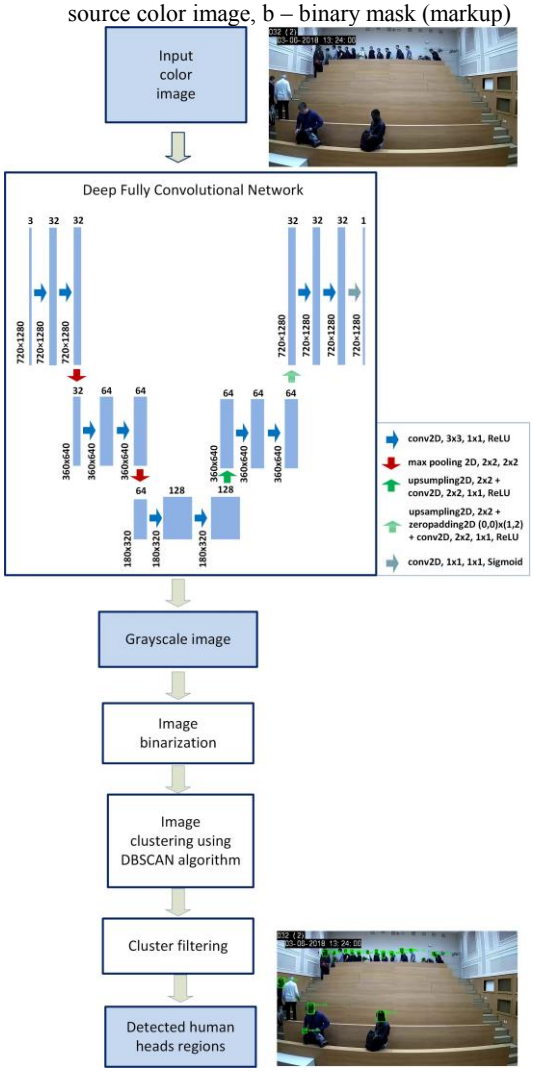


Figure 3. Human head detector based on FCN with clustering

4.2 Faster R-CNN architecture

During training, the network implementation with the Tensorflow object detection API was applied (Huang et al., 2017). Faster R-CNN is more precise detector than SSD (Liu, 2016) or YOLO (Redmon, 2015) architectures so they are not covered in this paper. Before being fed to the network input, the original color image was converted to a size of 1024×1024. Based on the masks we have generated markup corresponding to the format tf.record. Batch size is equal 1. Weights pre-trained in COCO dataset (COCO Consortium, 2018) were used for network initialization.

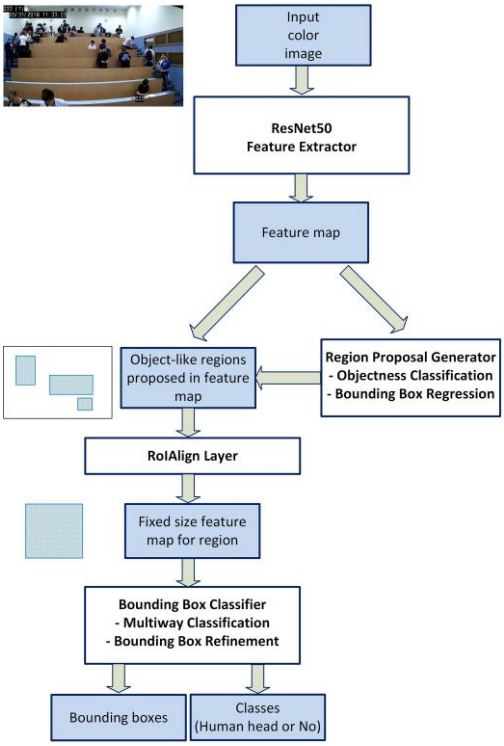


Figure 4. Human head detector based on Faster R-CNN

4.3 Mask R-CNN architecture

The Mask R-CNN model generates bounding boxes and segmentation masks for each object (human head) in the image. It's based on a ResNet101 backbone and Feature Pyramid Network (FPN) (Fig. 5).

Training process is described in detail by (Waleed, 2017). When applying to the input network the original image is converted to an image size of 1024 × 1024. Batch size is also 1.

Similarly, with the Faster R-CNN, the network is tuned using weights, pre-trained in COCO dataset.

The output is also supplemented with masks for each of the objects contained in the image. Information about masks allows us to make the network more accurate, because in addition to the bounding box of the object, we get its semantic segmentation. This allows filtering false positives of the network.

4.4 Quality of human heads detection using deep neural architectures

Table 1 shows a performance comparison of the three detectors based on deep convolutional neural networks.

The Mask R-CNN works more than ten times slower than the first one, but it almost does not give false positives and has the precision and recall of head detection over 90% on both test and training samples. The Faster R-CNN architecture gives worse accuracy than Mask R-CNN, but it gives fewer false positives than FCN with clustering.

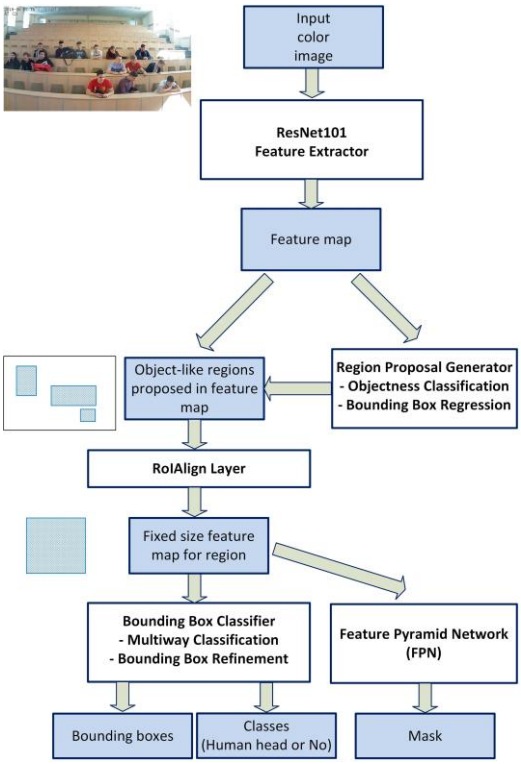


Figure 5. Human head detector based on Mask R-CNN

The calculations are performed on the graphics processor of the NVidia GTX1080 video card with support of CUDA technology.

Deep neural network architecture	Quality on training sample		Quality on test sample		Mean detection time, sec
	Precision	Recall	Precision	Recall	
FCN with clustering	0,8028	0,6687	0,7830	0,7464	**0,22**
Faster R-CNN	0,9572	0,9384	0,9243	0,8869	0,60
Mask R-CNN	**0,9976**	**0,9781**	**0,9953**	**0,9075**	2,91

Table 1. Comparison of deep neural architectures

Since the task formulation pays special attention to the quality of object detection and does not impose high demands on the computation speed, the Mask R-CNN architecture is chosen for further use as part of the software application.

5. SOFTWARE FOR THE DETECTION OF HUMAN HEADS USING DEEP LEARNING

5.1 Software implementation

Based on Mask R-CNN architecture authors have developed software for human head detection on a low-quality image, the structure of which is shown in Figure 6. It is a two-level web-service with client and server modules. This software is used to detect and count people in the premises.

The server module allows us to access video streams of a specified IP cameras list using the rtsp protocol, detect and count human heads using a trained R-CNN Mask neural

network, save recognition results to files and a database based on SQLite DBMS, and also generate a log-files with a history of events. Resolution of IP cameras is 1920×1080 pixels. Server hardware includes processor Intel Core i5-4570 3.2GHz with 4 Cores, 8 GB RAM, graphic card NVidia GeForce GTX1080 8Gb. Server operation system is Windows 7.

The server module is implemented in Python 3.5 using the vlc, pyqt5, keras, and django libraries. Apache is used as a web server. This solution is cross-platform and can function both under the Windows operating system and Linux.

Access to the client module is carried out from any computer connected to the local network of BSTU n.a. V.G. Shukhov by IP address of the server. Updating the results of people counting is done 1 time per 1 minute (can vary depending on the requirements). Client module is developed using Angular framework. When you click on a thumbnail room image the client module shows the result of image recognition by a neural network with the detected human heads and their count.

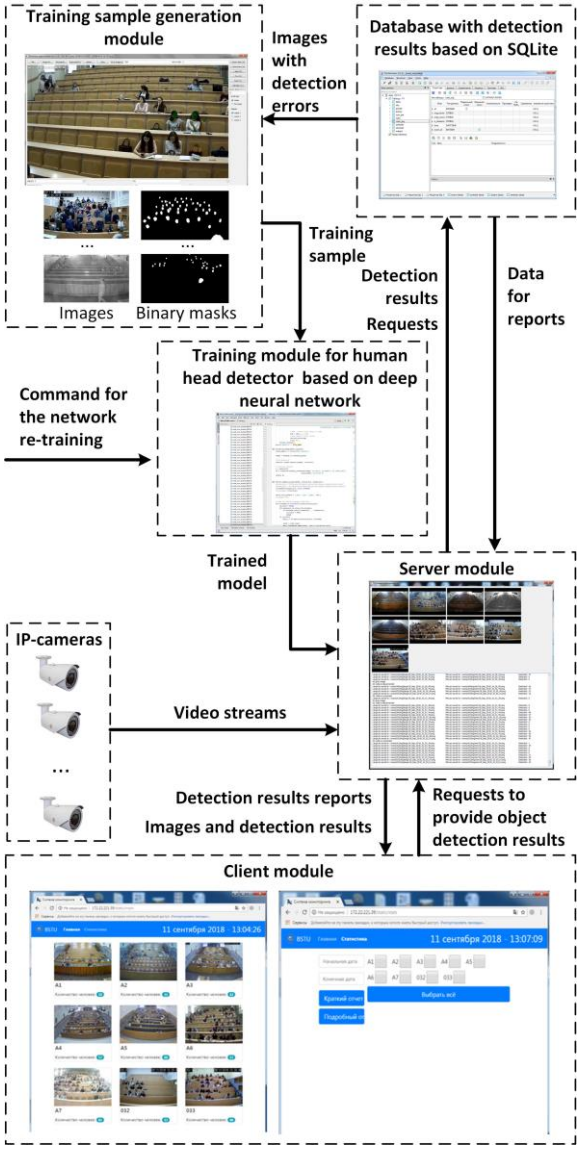


Figure 6. The structure of the developed software for human heads detection

5.2 Experimental results

The processing time of a single image of 1920×1080 pixels varies from 0.5 to 6 seconds depending on the number of people in the room. The mean absolute error of people counting in the room is 3,1 for Mask R-CNN. This indicates acceptable performance indicators of the developed software.

Examples of head detection in classrooms of BSTU n.a. V.G. Shukhov shown in Fig. 7. They demonstrate the fact of high quality of proposed neural network algorithm for objects detection under conditions of noise and when people are turning their backs to the video camera.

Figure 7. Examples of human heads detection in classrooms of BSTU n. a. V.G. Shukhov with developed software: a – people turned their backs to the camera, b – the image obtained under noise, c, d – detection results on images from university lectures, e – result on image with very different sizes of human heads

6. CONCLUSIONS

The test results show that the usage of deep convolutional neural networks allows us to reliably detect a human head on 2D images regardless of the turn to the observer. The Mask R-CNN architecture demonstrates high accuracy rates even on low-quality images, but imposes significant limitations on the speed of such algorithms. However, a large number of computer vision applications do not require real-time object recognition.
The developed software works with IP cameras, which ensures its scalability for detecting queues in buffets, visitors monitoring in retail, detecting pedestrians on the roads using outdoor video cameras, determining the workload of public transport stops, etc.

ACKNOWLEDGEMENTS

Research is carried out with the financial support of The Ministry of Education and Science of the Russian Federation within the Public contract project 2.1396.2017/4.6

REFERENCES

COCO Consortium, 2018. COCO Common Objects in Context. http://cocodataset.org

Dalal, N., Triggs, B., and Europe, D., 2005. Histograms of Oriented Gradients for Human Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1, pp. 886-893.

Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., and Zou, H., 2018. Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 145, Part A, pp. 3-22.

Ester, M., Kriegel, H. P., Sander, J., and Xu, X., 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, Portland, OR, AAAI Press, pp. 226-231.

He, K., Gkioxari, G., Dollar, P., and Girshick, R., 2017. Mask R-CNN. *arXiv:1703.06870v3*.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., and Murphy, K., 2017. Speed/accuracy trade-offs for modern convolutional object detectors. *CVPR 2017, arXiv:1611.10012v3*.

LeCun, Y., Bengio, Y., and Hinton, G., 2015. Deep learning. *Nature*, Vol. 521, pp. 436-444.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., 2016. SSD: Single Shot MultiBox Detector. *ECCV, arXiv:1512.02325*.

Olson, D. L., and Delen, D., 2008. Advanced Data Mining Techniques. Springer, 1st edition.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., 2015. You only look once: Unified, real-time object detection. *arXiv:1506.02640*.

Ren, S., He K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems 28 (NIPS 2015)*.

Ronneberger, O., Fischer, P., and Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Springer, LNCS, 9351, pp. 234–241.

Viola, P., Jones, M.J., and Snow, D., 2003. Detecting pedestrians using patterns of motion and appearance. *Proceedings of the 9 th International Conference on Computer Vision (ICCV'03)*, Vol. 1, pp. 734-741.

Waleed, A., 2017. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. *GitHub repository*, https://github.com/matterport/Mask_RCNN

Yudin, D., 2018. Segmentation tool with GUI based on pyQT5 and Opencv for Python 3. *Github Repository*, https://github.com/yuddim/ multi_class_segmentation_tool.

Yudin, D., and Slavioglo, D., 2018. Usage of Fully Convolutional Network with Clustering for Traffic Light Detection. *7th Mediterranean Conference on Embedded Computing, MECO'2018*, pp. 242-247.

Revised April 2019