

Design and Develop a One-Shot Learning Face Recognition System using Deep Convolutional Network

Md. Mehedi Hasan¹, Md. Jashim Uddin², Khandaker Takdir Ahmed³, Md. Rijoan Rabbi⁴, Md. Abdullah-Al-Imran⁵, Sonali Saha⁶

¹Assistant Computer Programmer, ICT Cell, Islamic University, Kushtia-7003, Bangladesh,

^{2,3}Assistant Professor, Dept. of ICT, Islamic University, Kushtia-7003, Bangladesh,

^{4,5}Student, Dept. of ICT, Islamic University, Kushtia-7003, Bangladesh,

⁶Instructor (Computer Technology), Magura Polytechnic Institute, Magura-7610, Bangladesh.

Abstract: Face recognition has recently received significant attention especially during the past few years at one of the most successful applications of image analysis and understanding. Facial recognition technology (FRT) has emerged as an attractive solution to address many contemporary needs for identity and verification of identity claims. The identification of humans by the unique characteristics of their faces is called face recognition. FRT technology is the least intrusive and fastest biometric technology. With the advancement in technology extracting information and within creasing security needs has become much simpler. The system proposed in this paper uses the power of the Convolutional Neural Network (CNN) to encode the face and produce a vector matrix. Then we use tripled loss function to calculate the distance between input and trained image to predict the face.

Keywords: Tensorflow, Face recognition, Keras, ImageNet, Convolutional Neural Network (CNN.)

Date of Submission: 12-05-2020

Date of Acceptance: 24-05-2020

I. Introduction

Recently face recognition is gaining much attention in the society of network multimedia information access. Areas such as content indexing, network security and retrieval, and video compression benefit from face recognition technology because people are the center of attention in a lot of video and pictures. Network access control via face recognition not only increases the user-friendliness in human-computer interaction. Indexing and retrieving video data based on the appearances of particular persons will be useful for users such as news reporters, political scientists, and moviegoers, but it also makes hackers virtually impossible to steal one's password[1]. For the applications of teleconferencing and videophone, the assistance of face recognition also provides a more efficient coding scheme.

The skill to learn object categories from a few examples, and at a rapid pace, has been demonstrated in humans. At the age of six of a child, it is estimated that it has learned almost all of the 10 ~ 30 thousand object categories in the world. This is due not only to its ability to synthesize and learn new object classes from existing information about different, previously learned classes but also to the human mind's computational power. Given two examples from two different object classes: one, an unknown, amorphous shape, the second, an unknown object composed of familiar shapes; it is much easier for humans to identify the former than the latter, suggesting that humans make use of existing knowledge of previously learned classes when learning new ones[2].

The motivation behind this project is that face detection has an amplitude of possible applications. Digital cameras automatically focus on a person's faces to security cameras that match a face to a person's identity. For locking a personal computer, webcams are often used as a security measure. The webcam's facial recognition technology allows for the computer to be accessible to a person only if it recognizes their face. This technology can be further narrowed down to the tracking and recognition of eyes.

Lately, a lot of work has been employed in Face Recognition. CNN's have a high computational cost in terms of memory and speed in the learning stage but can achieve some degree of shift and deformation invariance. Nowadays, this approach became more feasible thanks to the hardware evolution and the capability of using the GPU processors to perform convolutions and a large amount of available data that allows the learning of all CNN's parameters[3]. This network type has demonstrated being able to achieve high recognition rates in various image recognition tasks like character recognition, handwritten digit recognition; object detection, facial expression recognition, and face recognition. Although there are many methods in the literature, some aspects still deserve attention, for example, accuracy is somewhat low in and validation methods could be

improved and the recognition time could be a little improved to be performing in general. Moreover, we think to reduce computational cost and time as well as good accuracy.

Image files that contain human faces can be automatically recognized. This is one of the basic problems of computer vision and this is called Face recognition. Many problems in computer vision were facing problems with their accuracy before a decade[4]. However, with the update of deep learning techniques, the accuracy of these problems drastically improved. As we will demonstrate, convolutional neural networks are currently the state-of-the-art solution for Face recognition. The main task of this project is to develop and test a face recognition system for images based on convolution neural network and Siamese network one-shot learning.

In the theoretical part, we study relevant literature and how convolution neural networks improved the computer vision area in the past few years. And in the experimental part, we will show how easily a convolution neural network and triplet loss function can be implemented for face recognition in practice with faster test and high accuracy.

The main challenge of this project is workability with a lower amount of dataset. On the face recognition work, many other algorithms are used differently for many purposes but all of them are not much effective and efficient to recognize a person with a lower amount of dataset. We remove this barrier combining CNN with One-shot learning in this project.

The rest of this paper is organized as follows. Face recognition and one-shot learning are described in section II. Development tools and libraries are given in section III. Methodology and Implementation process is described in section IV. Finally, the Conclusion is given in section V.

II. Face recognition and One-Shot Learning

Scientists developed Facial-Recognition Software in the 1960s. The scientist's names were Woody Bledsoe, Helen Chan Wolf, and Charles Bisson.

One-shot learning is an object categorization problem. It is found mostly in computer vision. Whereas most machine learning-based object categorization algorithms require training on thousands of samples/images and very large datasets, one-shot learning aims to learn information about object categories from one, or only a few, training samples/images[5][6].

The main focus of this article will be on the solution to this problem presented by Fei-Fei Li, R. Fergus, and P. Perona in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol28(4), 2006, which uses a generative object category model and variational Bayesian framework for representation and learning of visual object categories from a handful of training examples. Another paper, presented at the International Conference on Computer Vision and Pattern Recognition (CVPR) 2000 by Erik Miller, Nicholas Matsakis, and Paul Viola will also be discussed[7].

III. Development tools and libraries

We used python programming language along with necessary development tools and different useful machine learning/deep learning libraries. Python is a great general-purpose programming language on its own, but with the help of a few popular libraries, it becomes a powerful environment for scientific computing. We choose python to build my model because python has many highly developed deep learning libraries which help me building this object detection model easily and more accurately. List of used python deep learning libraries and development tools are given below:

Python libraries for deep learning are Numpy, Tensorflow, Keras, Matplotlib. And the development tools are GoogleColab, Jupyter notebook.

IV. Methodology and Implementation

A. Methods

In this method, at first, we trained our model using a convolution neural network. We have used the Transfer learning model with a Happy house dataset collected from the Coursera sequence learning course. At first, we normalize the pixel values then we use these values to fit the transfer learning model. After that, we feed the images to the deep architecture CNN to features extraction process to learn the parameters from the images which also called the encoding process. In the second step, we feed the output of the CNN model as a 128-dimensional vector. The vector space learns with the label from the given images in the database. Finally, we get the comparison result of those one-hot vectors and get the minimum distance and match the authorized person from the database which contains a lower amount of user face data[8][9].

Here, we use a deep convolutional network and an embedded triplet loss function. We discuss two different core architectures:

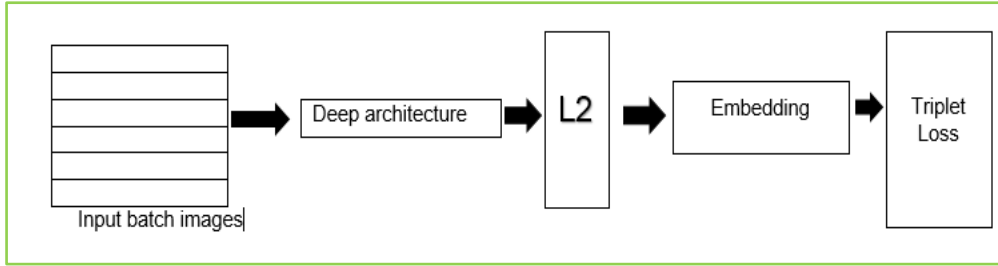


Figure 4.1: Model structure of Our network

B. Triplet Loss

The embedding is represented by $\text{by}f(x) \in \mathbf{R}^D$ it embeds an image x into a d -dimensional Euclidian space. Additionally, we constrain this embedding to live on the d - dimensional hypersphere, that is $\|f(x)\|_2 = 1$. This loss is motivated by the context of nearest-neighbor classification. Here we want to ensure that an image x_i^a (anchor)of a specific person is closer to all other images x_i^p (positive) of the same person than it is to any image x_i^n (negative)of any other person. This is visualized in Figure 3.

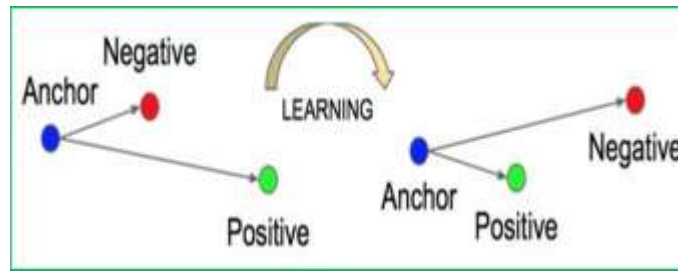


Figure 4.2: The Triplet Loss direction

$$\text{Thus we want } \|f(x_i^a) - (x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - (x_i^n)\|_2^2 \dots\dots\dots(1)$$

$$\forall, f(x_i^p), f(x_i^n) \in T \dots\dots\dots(2)$$

where α is a margin that is enforced between positive and negative pairs. T is the set of all possible α triplets in the training set and has cardinality N . The loss that is being minimized is then

$$\sum_i^N [\|f(x_i^a) - (x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - (x_i^n)\|_2^2 + \alpha] \dots\dots\dots(3)$$

Generating all possible triplets would result in many triplets that are easily fulfilled the constraint in Eq. (1)[9]. These triplets would not contribute to the result in slower convergence, as they would still be passed through the network. It is crucial to select hard triplets that are active and can, therefore, contribute to developing the model.

C. Triplet Selection

In order to ensure fast convergence, it is crucial to identify triplets that violate the triplet constraint in Eq. (1). This means that, given x_i^a , we want to select an x_i^p (hard positive) such that $\text{argmax } x_i^p \|f(x_i^a) - (x_i^p)\|_2^2$ similarly x_i^n (hard negative) such that $\text{argmin } x_i^n \|f(x_i^a) - (x_i^n)\|_2^2$.

It is infeasible to calculate the argmin and argmax across the whole training set. Additionally, it might lead to poor training, as mislabeled and poorly imaged faces would dominate the hard positives and negatives. Two choices avoid this issue:

- Generate triplets offline each n steps, using the most recent network checkpoint and computing the argmin and argmax on a subset of the data.
- Generate triplets online. This can be done by selecting the hard positive or negative exemplars from within a mini-batch.

Selecting the toughest negatives can in practice lead to bad local minima early on in training, specifically, it can result in a collapsed model (i.e. $f(x)=0$). For mitigating this, it helps to select x_i^n such that

$$\|f(x_i^a) - (x_i^p)\|_2^2 < \|f(x_i^a) - (x_i^n)\|_2^2 \dots\dots\dots(4)$$

We call these negative exemplars semi-hard, as they are further away from the anchor than the positive exemplar, but still hard because the squared distance is close to the anchor positive distance. Those negatives lie inside the margin α [11].

D. Datasets and Evaluation

We evaluate our method happy house datasets and with the exception of Labelled Faces in the Wild and YouTube Faces we evaluate our method on the face verification task. That is, given a pair of two face images a squared L2 distance threshold $D(x_i, x_j)$ is used to determine the classification of the same and different. All faces pairs (i; j) of the same identity are denoted with P_{same} whereas all pairs of different identities are denoted with P_{diff} . We define the set of all true subjects as

$$TA(d) = \{(i, j) \in P_{same}, with D(x_i, x_j) \leq d\} \dots\dots\dots(5)$$

These are the face couple (I, j) that were correctly classified as same at threshold d. Similarly

$$FA(d) = \{(i, j) \in P_{diff}, with D(x_i, x_j) \leq d\} \dots\dots\dots(6)$$

is the set of all couples that were incorrectly classified as the same (false accept). The false accept rate FAR(d) and the validation rate VAL(d) for a given face distance d are then defined as

$$VAL(d) = \frac{|TA(d)|}{|P_{same}|} \text{ and } FAR(d) = \frac{|FA(d)|}{|P_{diff}|} \dots\dots\dots(7)$$



Figure 4.3: Happy House dataset from Coursera.org

E. Project Implementation:

We train the image dataset representations via a supervised metric-based approach with siamese neural networks, then reuse that network’s features for one-shot learning. To develop a model for one-shot image classification, we aim to first construct a neural network that can discriminate between the class-identity of image pairs, which is the standard verification task for image recognition. The verification model learns to identify input pairs according to the probability that they belong to the different classes or the same class. This model can then be used to calculate new images, exactly one per novel class, in a pairwise manner against the test image. The pairing with the peak score according to the verification network is then awarded the highest probability for the one-shot task[12]. If the features learned by the verification model are sufficient to confirm or deny the identity of each person from one set of few images (it’s maybe ten or less), then they ought to be sufficient for other person’s identity, provided that the model has been exposed to a variety of persons to encourage variance amongst the learned features.

F. One-shot Learning:

One-shot learning is the technique of learning representations from a sample. Take the example previously mentioned, suppose there is an organization and it wants a facial recognition system to allow access to the building for its employees and you are given the task of building such a system. The problem with this task is that the organization might not have more than ten images for each of the employees. Therefore, building and training a typical convolutional neural network will not work as it cannot learn the attributes required with the given amount of data. So, this is a one-shot learning task where you build a similarity function that compares two images and says you if there is a match[13].

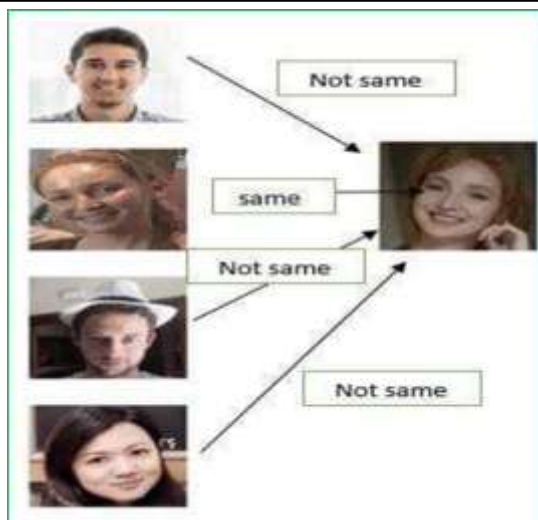


Figure 4.4: Face matching by One-shot learning

Suppose, the images on the left are faces of employees of the organization. Since there isn't much data to create a CNN, you can build a similarity function that compares the image on the right with all the images on the left. The similarity function will provide value and if that value is lesser than or equal to a threshold value, you could say that the two images are similar, else they are not.

G. Siamese Network:

In Siamese networks, we can give an input image of a person and find out the encodings of that image, then, we take the same network without performing any updates on weights or biases and input an image of a dissimilar person and again predict it's encodings.

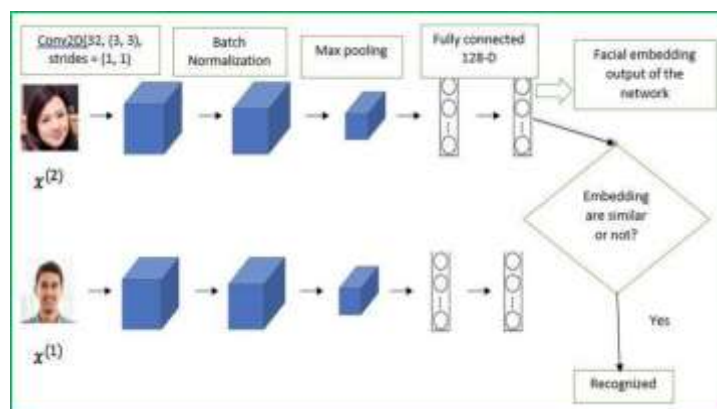


Figure 4.5: Project architecture with Siamese Network

Now, we compare these two encodings to estimate whether there is a similarity between the two images. These two encodings act as a latent attribute representation of the images. Images with the same person have similar features/encodings. Using this, we compare and tell if the two images have the same person or not. When the images with labeled are less than the threshold distance we consider that as the same person if the distance among them is greater than the threshold level we consider that mislabel with the person's face called as an unauthorized person.

H. Face Clustering:

This method employs as the semi-supervised learning where some photos are labeled and some are may non labeled so this algorithm work as the clustering techniques. Our compact embedding lends itself to be used in order to cluster user's photos into groups of people with the same identity label. The constraints in assignment imposed by clustering faces, compared to the pure verification task lead to truly amazing results.

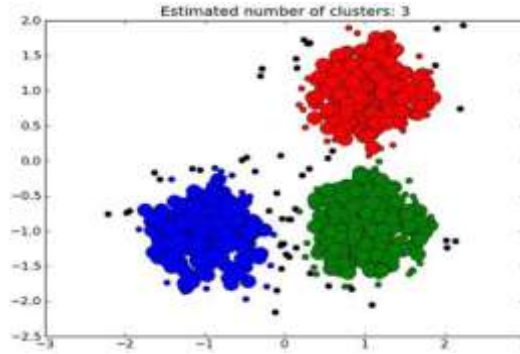


Figure 4.6: Clustering dataset

Figure 4.6 shows one cluster in a user’s photo collection, generated using agglomerative clustering. It is a clear showcase of the incredible invariance to occlusion, lighting, and even age.

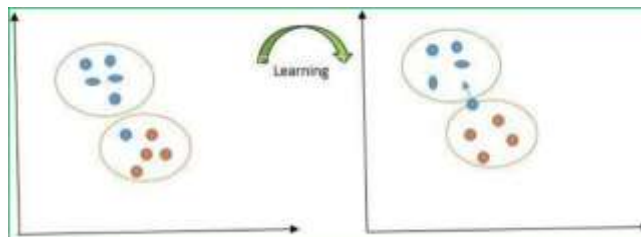


Figure 4.7: Improvement of learning steps

On figure 4.7 shows that if some picture is different in pixel value in the same label are also can understand as the same person image by learning with the label.

I. Triplet loss for verification:

We train the network by taking an anchor image and comparing it with both a positive sample and a negative sample. The dissimilarity between the positive image and the anchor image must low and the dissimilarity between the anchor image and the negative image must be high

$$\|f(x_i^a) - (x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - (x_i^n)\|_2^2 \dots\dots\dots(8)$$

The formula above represents the triplet loss function using which gradients are calculated. The variable “a” represents the anchor image, “p” represents a positive image, and “n” represents a negative image[13].



Figure 4.8: Triplet loss Verification

We know that the dissimilarity between a and p should be less than the dissimilarity between a and n. Another variable called margin, which is a hyperparameter is added to the loss equation. Margin defines how far away the dissimilarities should be, i.e if margin = 0.7 and d(a,p) = 0.02 then d(a,n) should at least be greater than 0.028. Margin helps us distinguish between the two images better.

Therefore, by using this loss function we evaluate the gradients and with the help of the gradients, we update the weights and biases of the siamese network. For training the network, we take an anchor image and randomly negative images and sample positive and compute its loss function and update its gradients.

J. Results:

Our model is shown more effective and an emerging result for the low amount of dataset we use because of the use of one hot vector in the learning parameter. The approaches we used to recognize the faces from the database which approximately gets 56% accuracy in the test set image with a tiny training time to calculation. Less amount of time is possible because we use only use one layer CNN architecture to train our data. By the use of One hot vector to measure the distance between the anchor image and test image. We get only 81297 parameters in our train model. Our model shows output at a satisfying level which can recognize the person from several angles and the variation of light intensity hasn't much effect on the output of the result. The output of our model mainly varies with the threshold level of the distance matrices. In our model, we use .028 which also depends on the value of we use this value as 0.28. The number of layers and the size and number of filters we use the parameters to improve the accuracy and effectiveness in our model. The more layer we use the more accurate output we will get but the time and space(memory) trade-off is the main curse in this method with a huge number of parameters. To overcome the overfitting of the model we use the Dropout function after the convolution operation that makes our result more effective and efficient.







Database: Rijoan	
 Train: 1.jpg	 Test: 2.jpg Output: It's Rijoan, Vector distance: 0.068
Database: kian	
 Train: kian.jpg	 Test: camera_2.jpg Output: It's not kian, Vector distance: 0.03338
Database: Mehedi	
 Train: Mehedi.jpg	 Test: mtest.jpg Output: It's Mehedi, Vector distance: 0.02652

Figure 4.9: Three Databases namely Rijoan, Kian, Mehedi and their respected Output

Put more images of every person (under different lighting conditions, taken on different days, etc.) into the database. Then given a new image, differentiate the new face to multiple pictures of the person. This would improve accuracy. Select the images to just contain the face and less of the border around the face. This preprocessing deletes some of the irrelevant pixels around the face, and also makes the algorithm more robust.

Face verification clears up an easier 1:1 matching problem; face recognition addresses a harder 1:K matching problem. The triplet loss is a productive loss function for training a neural network to learn an encoding of a face image. The same encoding can be used for recognition and verification. Calculating distances between two images' encodings allows you to determine whether they are pictures of the same person.

V. Conclusion

The aim is reached by face detection and recognition methods. Knowledge-Based face detection methods are used to calculate, locate, and extract faces in acquired images. Implemented methods are skin color and facial attributes. The neural network is used for face recognition and detection. RGB color space is used to identify skin color values, and segmentation decreases the searching time of face images. Facial components on face candidates seem with the implementation of the LoG filter. LoG filter identifies good performance on extracting facial components under different illumination conditions.

Acknowledgments

This work was supported by Dept. of ICT, Islamic University, Kushtia-7003, Bangladesh.

References

- [1]. Sun, Yi, Xiaogang Wang, and Xiaoou Tang. "Hybrid Deep Learning for Face Verification." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.10 (2016): 1997-2009
- [2]. Hu, Guosheng, et al. "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition." *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015.
- [3]. Zhang, Tong, et al. "A deep neural network driven feature learning method for multi view facial expression recognition." *IEEE Trans.Multimed*99(2016): 1
- [4]. Florian Schroff, Dmitry Kalenichenko, James Philbin (2015). "FaceNet: A Unified Embedding for Face Recognition and Clustering"
- [5]. Yaniv Taigman, Ming Yang, Marc'AurelioRanzato, Lior Wolf (2014). "DeepFace: Closing the gap to human-level performance in face verification"
- [6]. The pretrained model we use is inspired by Victor Sy Wang's implementation and was loaded using his code: <https://github.com/iwantooxxoox/Keras-OpenFace>.
- [7]. Our implementation also took a lot of inspiration from the official FaceNetgithub repository: <https://github.com/davidsandberg/facenet>
- [8]. Kanade, Takeo. "Picture processing system by computer complex and recognition of human faces" *Doctoral dissertation*, Kyoto University 3952(1973): 83-97.
- [9]. Brunelli, Roberto, and Tomaso Poggio. "Face recognition: Features versus templates." *IEEE transactions on pattern analysis and machine intelligence* 15.10(1993): 1042-1052.
- [10]. D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol.60, no.2, pp.91–110, Nov.2004.
- [11]. Biggio, B., Fumera, G., & Amp, Roli, F. (2014). "Pattern recognition systems under attack: Design issues and research challenges". *International Journal of Pattern Recognition and Artificial Intelligence*, 28(07), 1460002
- [12]. B. Klare and A. K. Jain, (2010) "On a taxonomy of facial features," in *Proc. 4th IEEE Int. Conf. Biometrics Theory, Applications and Systems (BTAS)*, Crystal City, Washington D.C.
- [13]. Leonardo A. Camenta, Francisco J. Galdamesa, Kevin W. Bowyer, Claudio A. Perez,(2015) "Face Recognition under Pose Variation with Local Gabor Features Enhanced by Active Shape and Statistical Models"
- [14]. P. Karthigayani, S. Sridhar,(2014) "Decision tree based occlusion detection in face recognition and estimation of human age using back propagational network"*Journal of Computer Science*.
- [15]. Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. *Imagenet classification with deep convolutional neural networks*. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Md. Jashim Uddin, et. al. "Design and Develop a One-Shot Learning Face Recognition System using Deep Convolutional Network." *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, 15(3), (2020): pp. 16-24.