***M. Shtyl**, BA student*
***M. Ukrainets**, lecturer*
***S. Kukharonok**, senior lecturer*
Zhytomyr Polytechnic State University

DEVELOPMENT OF A PIPELINE AND A FACE RECOGNITION MODEL BASED ON CONVOLUTIONAL NEURAL NETWORKS AND METRIC LEARNING

The development of computer vision systems is one of the most pressing areas of AI algorithm development. The creation of accurate facial recognition algorithms is the primary driver of demand for these technologies. Traditional methods often prove insufficiently robust against changes in lighting or camera angles. The current standard for solving this problem is the use of deep learning methods. The aim of this research is to study these technologies and develop an optimized neural network pipeline capable of extracting human features and operating effectively in verification (1:1) and identification (1:N) scenarios.

To ensure inference stability and high recognition accuracy, an image preprocessing algorithm has been developed.

Using keypoint analysis (Face Mesh), the coordinates of the pupil centers are determined, after which an affine transformation is performed to correct for tilt and scaling. The face is positioned so that the eyes are at a clearly fixed level. The final size of the input image is reduced to 96x112 pixels. Thanks to this, the neural network does not waste resources on compensating for spatial distortions.

A convolutional neural network (CNN), specifically the ResNet18 architecture, was chosen as the backbone [1]. The use of residual connections effectively addresses the problem of gradient vanishing.

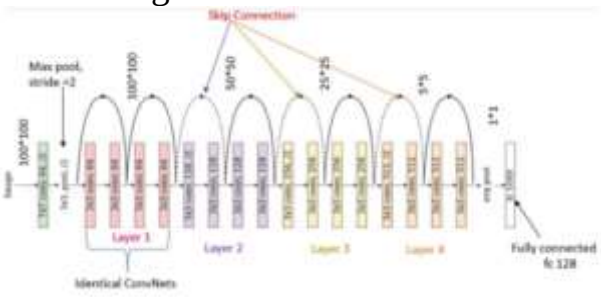

Fig. 1 - Feature extractor architecture

This architecture strikes an optimal balance between the depth required to recognize complex facial patterns and the overall computational complexity of generating a 512-dimensional feature vector.

To achieve high resolution between different classes, the ArcFace loss function is used. This algorithm projects vectors onto a hypersphere and adds an angular penalty directly to the cosine of the angle between the feature vector and the target class weights, thereby maximizing the inter-class distance.

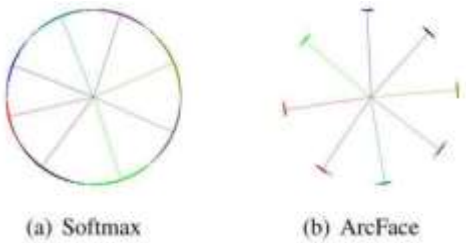

Fig. 2 - Visualization of the distribution of feature space using the ArcFace function

The model was trained using an RTX 4070 graphics processing unit (GPU), supported by CUDA toolkit for accelerated AI computations [1]. The CASIA-WebFace dataset, containing over 478,000 images, was used. Dynamic hyperparameter tuning was implemented to stabilize the gradient descent process. Specifically, a custom scheduler was used to incrementally increase the penalty, linearly raising the ArcFace angular penalty from its minimum value to the target value (0.35) during the first few epochs. This allows the network to first learn basic facial features without the risk of instability.

In addition, a mechanism for delayed learning rate reduction was integrated: adaptive learning rate reduction is activated only after the angular penalty has fully stabilized at its maximum value. To optimize memory usage, accelerate computations, and efficiently utilize GPU resources, mixed precision was employed [2].

The trained model was evaluated by calculating the cosine similarity between the generated normalized feature vectors. An optimal cutoff threshold of 0.45 was determined experimentally.

Testing on photos of real people in 1-to-1 scenarios confirmed that the system successfully verifies a person even in the presence of changes in appearance, such as the growth of a beard, and from different camera angles. This indicates that the model has learned to recognize the deep structure of the face while ignoring background noise.

Analysis of loss metrics during training demonstrated the absence of critical overfitting, confirming the model's high generalization ability [3].

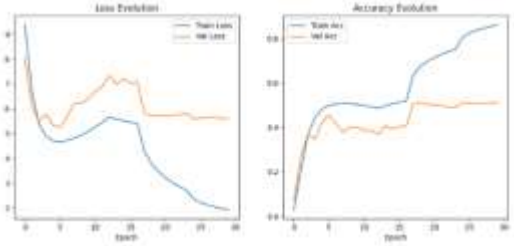

Fig. 3 - Metrics over the course of training: Loss and Accuracy plots

Fig. 4 – Testing on real data

The proposed method for developing a model based on the ResNet18 architecture and the ArcFace function demonstrates practical effectiveness. The application of keypoint-based pre-alignment algorithms and the integration of custom

hyperparameter schedulers made it possible to develop an accurate and stable feature extractor suitable for use in modern biometric systems.

REFERENCES

1. Jason Brownlee. Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python – Machine Learning Mastery, 2019. – 563 c.
2. Torralba, P. Isola, W. Freeman. Foundations of Computer Vision (Adaptive Computation and Machine Learning series) – The MIT Press, 2024. – 840 c.
3. Brownlee J. How to Reduce Overfitting in Deep Learning with Weight Regularization [Online resource]. – Available at: https://machinelearningmastery.com/how-to-reduce-overfitting-in-deep-learning-with-weight-regularization/