Personal Identification Based Automatic Face Annotation in Multi-View Systems

Plamen Hristov, Ognian Boumbarov and Krasimir Vachev
Radiocommunications and Videotechnologies Department, Faculty of Telecommunications
Technical University of Sofia
8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria
{plm, olb, krasvachev}@tu-sofia.bg

Abstract – **We propose a method for automatic face annotation in a closed multi-view environment. In such an environment faces are automatically detected and images are collected from several sources. Then, for the purpose of real-time labelling, a model is incrementally trained with every new person who enters the environment. The incremental learning model is validated on the Pandora dataset, which is split into different classes for each incremental step. Active learning is used to determine the type of images needed for training based on a predefined budget.**

Keywords – **incremental learning; active learning; automatic face identification; multi-view systems;**

I. Introduction

The last decade has been a time of significant innovation in the area of deep learning and artificial intelligence. Easily available data and computing power, as well as high predictive capabilities have helped deep neural networks to overtake classical methods as a preferred type of model in any area of pattern recognition. Many milestones have been reached in areas like face detection, face recognition and face pose estimation [1][2][13][15][17][18]. Those tasks are integral to real-time face identification systems, where face images are continuously received and trained upon.

Unfortunately, some common problems in such systems exist, such as face occlusion, facial expression or face pose changes, all of which are characteristic of single view applications. Adding other views would aid significantly in overcoming those problems by giving new perspectives for every data sample. For example, a multi-view implementation could combine several models for each view and achieve more accurate final score either by score-fusion at the last layer, or by combination of extracted features at the earlier layers [20]. Even though this approach sounds simple and plausible, it needs to be modified in order to deal with events, specific to real-time environments, like new people entering the system or other reappearing. The end goal is to optimally conserve the information for those people without potential performance degradation.

Because most classification methods after compilation have immutable parameters like input and output sizes, they have to be retrained every time with a new class of samples. This would put a severe performance limitation in an online system in terms of computational speed and memory. Therefore, updating the trained model with new examples would be necessary. This process is called incremental learning [16]. Existing methods, implementing it, aim to utilize or modify existing knowledge in an optimal manner. A lot of them concentrate on experiments with benchmark datasets, where they find how the accuracies for different classes of data change with the introduction of new data [7] [8][9]. Applications, using facial images, are relatively rare in literature, which is partly the motivation for our paper.

Another problem occurs when a model of any kind has to be trained with a huge amount of data. This could consume a lot of time, which in many cases cannot be afford in real-time systems. That is why samples should be selected for training based on their informativeness or representativeness. This is a type of active learning - an approach that aims to use the least amount of data to achieve the highest possible model performance. It is most often used in situations with a lot of unlabeled data that has to be selected for labelling. The labelling is done by an oracle, which could be a user or a predictive model. The selection is done by a learner module.

Combining active learning with incremental learning would increase the performance of a real-time system. However, both tasks are mutually conflicting because of the assumption that all samples are labelled beforehand. Labelling new data while incrementally training a model could negatively affect the final performance of the model owing to the fact that the oracle would give a lot of falsely classified samples in the initial phase of its training. The proposed method assumes that the capturing devices with a shared field of view are calibrated in a way that each present person's position could be matched between views. Thus, it could be said that we always know who the corresponding person is for every captured frame at any given moment. In future experiments a system could be developed where people are labelled in real-time, which would be another challenge to solve.

In our method we have chosen the face pose as a selectable characteristic for the active learning task. This is done under the premise that the face pose is the most discriminative characteristic in a face image. The face pose refers to the direction of a human face with respect to its coordinate system in a three-dimensional space. Such three-dimensional space is described by the three Euler angles – yaw, pitch and roll, which define the rotation around each axis. (Fig. 1)

Another type of selection is the one which is based on extracted feature maps, which could be influenced by secondary attributes, such as the lightning conditions, the scale factor in the image, accessories like glasses and hats, facial expression, hair, etc. All of these characteristics are more specific to real-world scenarios.

978-1-6654-9878-4/22/$31.00 ©2022 IEEE

In summary, the contributions of our work are:

- We transfer a generative network to the field of face recognition and test its capabilities on a known dataset.
- We implement a novel incremental learning method for this network by utilizing generated images for pseudo-rehearsal
- We further improve the learning method by adding a module based on active learning.

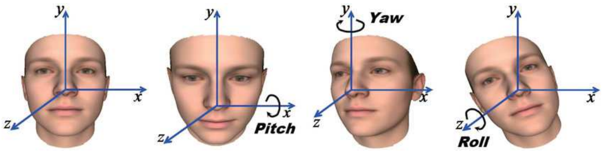

Fig. 1. Different illustrated face poses with a yaw/pitch/roll coordinate system, illustrated by [1]

II. Related works

Incremental learning is not a novel idea and different works have been proposed over several decades. Sometimes in the learning process of a network when the training set is too large, all the data cannot be loaded into the memory. If the quantity of data is not enough the accuracy of the model would decrease. Therefore, a way to divide the dataset into pieces so that the training process is effective needs to be found.

Another problem occurs when new classes have to be added in the learning process. If the network should be trained again with the whole data, vital resources and time would be lost. Often when new classes are added, the accuracy of the network decreases. On the other hand, the network's parameters increase. The conclusion is that there has to be a way to overcome these drawbacks and to make the training process stable in the described conditions – adding new classes in the process. Incremental learning has the potential to solve these problems.

Syed et. al [6] is a notable example for implementing such an approach for the Support Vector Machine (SVM). Due to how SVMs work, they have to be retrained for new classes. The authors prove that the models could be trained efficiently by taking the calculated support vectors as samples, representing old classes. This significantly reduces training time and memory to preserve the old data.

Recent incremental learning methods are based on deep learning models. The main problem they usually concentrate on is solving the catastrophic forgetting, which characterize such models.

Castro et. al. [7] propose a deep model, trained with a loss function, combined from cross-entropy and distillation loss, which helps to retain knowledge from old classes. The authors investigated two memory setups in order to find the most effective way how to keep the accuracy high, while adding new classes into the network. There are four main stages used in the paper for the incremental method. First, the data is collected. Then the network is trained. After that a fine-tuning is performed with the training set. Last, the memory is updated to include samples from new classes.

Kemker et. al. [8] developed a generative model called FearNet which does not store previous samples in the memory. With short-term memory for recent training examples and Deep Neural Network (DNN) for the older examples, the network obtains vital information and overcomes the catastrophic forgetting.

Xiang et. al. [9] utilize a generative model to generate exemplars from old classes. Their method implements incremental learning by calculating the mean and covariance matrix of feature vectors, learned from examples of old classes. Those statistics are later sampled from. Then the samples are used for fine-tuning a classifier.

Active learning has become increasingly popular as a means for minimizing the labeling costs in various fields of deep learning, where a lot of data is handled. Recent approaches are categorized into two main types: uncertainty-based and diversity-based.

Kim et. al. [10] propose a core-set approach for active learning, which uses diversity sampling by maximizing the distance between examples. They optimize this approach by estimating the density from new samples and select those that are in sparse regions.

Belouadah et. al. [12] apply a two-phase sample acquisition process to the incremental learning problem. They introduce two balancing acquisition functions, which select samples that are closer to minority classes.

Weinstein et. al. [19] use selective sampling with Minimal Margin Scores (MMS). At each training step they pass a batch of data to the model and select a small subset from this batch, which has the smallest MMS.

Real-time video systems are divided into two types with relation to the number of views – single-view and multi-view.

Ullah et. al [4] propose a method for automatic face recognition in CCTV systems. They collect a large dataset of 90 people, captured by a 15 fps camera, totaling more than 40000 frames. Each frame is converted to grayscale, and enhanced through edge detection. Then, the face in it is detected by the Viola-Jones algorithm and cropped. All of the face images are represented as eigenvectors through principal component analysis (PCA) and passed to a Convolutional Neural Network (CNN).

Ye et. al [11] present an approach for person tracking and re-identification in an indoor system with multiple IP cameras. People are detected and tracked using different deep CNN and their identity is matched between frames using the Hungarian algorithm. Personal identification is done using feature vectors, extracted from the CNN and calculating the distance between them and those already saved in the database.

Another example of a closed-door multi-view system is presented in [5]. It is an enclosed hexagonal space, called "Bee-Cube", where six depth sensors Kinect v2 are situated at equal distances, capturing a shared view. Each sensor is connected to a single-board computer, which establishes a connection with a central server, where the data is sent and processed. Face images are collected in real time by the Kinect v2 SDK, processed by PCA and classified with a SVM.

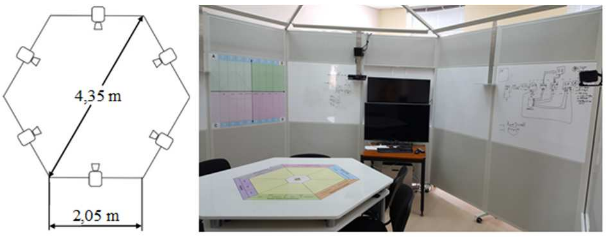

Fig. 2. Proposed “Bee-Cube” multi-view system in [5]

III. Proposed method

The overall method is based on the Auxiliary Classifier GAN (AC-GAN), which is a type of Conditional GAN, with an additional supervised output.

A. Generative Adversarial Network (GAN)

A GAN is a type of a machine learning architecture, where two models – a generator and a discriminator, compete in a zero-sum manner in the form of min-max optimization:

$$\min_{G} \max_{D} V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_{x}(z)}\left[\log\left(1 - D\big(G(z)\big)\right)\right] \quad (1)$$

where D is the discriminator, G is the generator, $p_{data}(x)$ is the probability distribution of real data, $p_x(z)$ is the probability distribution of fake data and z are the Gaussian latent vectors. Typically it is used for generating fake, but plausible data after learning the patterns and regularities of its real counterparts in the training process.

Generally, when referring to GANs, Deep Convolutional GAN are implied (Fig. 3). Here, the generator's role is sampling vectors randomly from a Gaussian distribution, reshaping them into tensors and up-scaling the tensors into images through the use of transposed convolution layers. Those images are used to deceive the discriminator, which tries to classify them into real or fake. At the same time, the discriminator also receives real images. The feedback from this classification is used to train the generator by means of adversarial loss, which ideally should be minimized. The discriminator is separately optimized based on its correctly or incorrectly classified samples through the same loss, which the generator is trying to maximize.

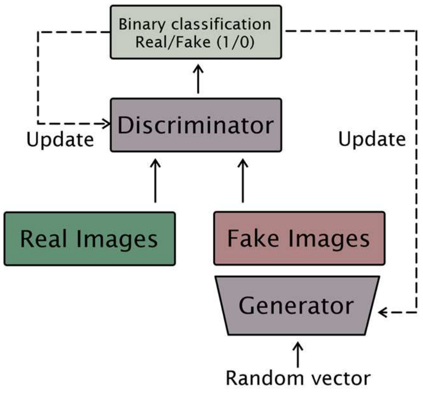

Fig. 3. Basic GAN architecture.

B. Auxiliary Classifier GAN for Incremental Learning

Conditional GAN models concatenate a feature vector, representing a class of data, to the random latent vector. This lets the user control the type of data that is being generated, based on its class. The AC-GAN structure has an additional supervised output from the discriminator for the class of the data. The cross-entropy loss from this output is added to the adversarial loss for optimizing the discriminator, which further stabilizes the training and in our case is used for the incremental learning part. The formula for the loss is:

$$L_{CE} = -\sum_{i=1}^{n} t_i \log(p_i) \quad (2)$$

where t_i is the ground truth label, n is the number of classes and p_i is the softmax probability for the i-th class.

The class feature vectors, which are input to the generator, are extracted from the second last layer of a ResNet-50[14] model, trained on the VGGFace2 dataset[15]. Compared to the basic class embeddings, which consist of raw or encoded class labels, the feature vectors represent a higher level of abstraction.

The full architecture of the AC-GAN is shown on Fig. 4.

It consists of a base-net, which acts as a feature extractor, a generator, and a discriminator. New images are passed to the base-net model, with their feature vectors extracted from the dense layer and passed to the generator.

The generator comprises 4 subsequent blocks of transposed 2D convolution layers, and batch normalization layers. The input vector is reshaped into a squared image, which gets up scaled doubly after each block. The discriminator has 4 convolutional layers, with dropout layers of 0.5 in-between. At the end, the feature maps are flattened and passed to a dense layer, which is connected to the two output dense layers for the classification and adversarial training. This connecting dense layer, as well as the base-net dense layer, has a hyperbolic tangent activation and a size of 256.

All (transposed) convolutional layers are activated by Leaky ReLU activation with a slope of 0.2.

C. Incremental Learning

Let L be a sequence of image subsets $\{X_0, X_1, X_2, \dots X_N\}$ for N incremental steps. Each batch X is a tensor with a size of $M \times W \times H \times C$, where M is the batch size, W and H are the width and height of the images respectively, and C is the number of channels (3 for RGB). The subsets contain a fixed amount of classes, with all of their corresponding samples, and do reappear in subsequent subsets.

The training of the GAN is done in batches. For every batch of real images, a batch of fake images with same size is generated from the feature vectors, extracted from the real images. A new GAN is retrained for every incremental step. Its generator is then used in future steps to generate images of old data when training the discriminator as a classifier. The number of generated images is equal to the number of new images.

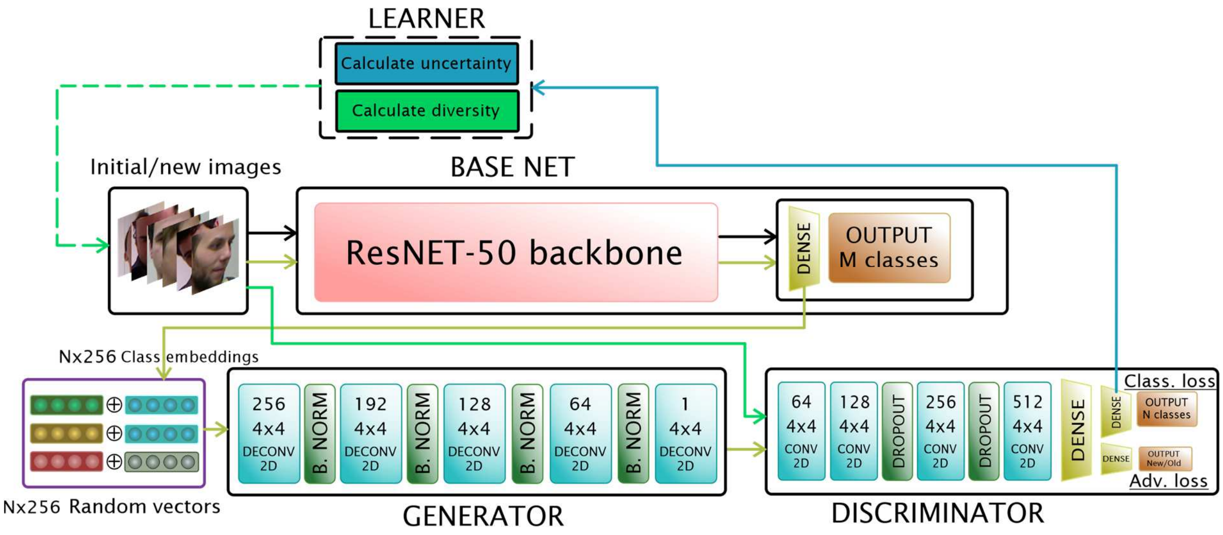

Fig. 4. Our GAN architecture, based on AC-GAN. The black arrows define the preliminary training of the base-net for future feature extraction. The green and yellow arrows show the adversarial training for the discriminator and generator, respectively

The full algorithm for incremental learning is as follows:

1. *Prepare sequence of incremental sets of data* $\{X_0, X_1, X_2, \dots X_N\}$
2. *Build AC-GAN model* $A = \{B; G; D\}$
3. *For first incremental step/set:*
   a. *Train G and D in an adversarial manner with the loss function from Eq.1 and Eq.2;*
4. *For each subsequent incremental set* ***n****:*
   a. *Replace the supervised output layer of the generator with a one that fits all old and new classes.*
   b. *Detach the discriminator with the supervised output only.*
   c. *Train the detached discriminator with images from new classes and generated images from the last generator.*
   d. *Train G and D in an adversarial manner with the loss function from Eq.1 and Eq.2 on the new data only*

D. Active Learning Module

Another contribution to the existing method is configuring it for automatic face annotation. In addition to the existing modules of the GAN model, a learner module is added.

Based on a predefined budget of samples for each incremental batch, the learner selects the examples which give the highest entropy. This is calculated from the classification loss (Eq. 2) of the discriminator output. The images are sampled uniformly along their yaw/pitch/roll angles, which are quantized into groups of predefined size. Another way to look at this is by defining a 3D histogram for the angles, which should have equal values. (Fig. 5)

The algorithm for active learning is as follows:

1. *Define incremental sets* $\{X_0, X_1, X_2, \dots X_N\}$*, total budget* $\boldsymbol{O}$*, batch size* $\boldsymbol{B}$
2. *For every incremental set*
   a. *Define empty 3D histogram* $\boldsymbol{H}$ *for every class*
   b. *For every batch* ***b*** *in current incremental step:*
      - *For first batch:*
        - *Sample randomly from each class subset with a sample size of* ***B****, divided by the number of classes*
      - *For every subsequent batch:*
        - *Allocate sample size selection for each class, based on the ratio of the cross-entropies from the last batch*
        - *Sort the quadrants of the histogram* $\boldsymbol{H}$ *for every class by sample frequency*
        - *Select uniformly from each quadrant, starting from the most empty*
      - *Train the model on the sampled batch*
      - *If total sampled examples exceed* $\boldsymbol{O}$ *– quit learning process*
      - *Calculate cross-entropy for every class using (2)*
      - *Fill histogram* $\boldsymbol{H}$ *for every class, based on the pose of the images*

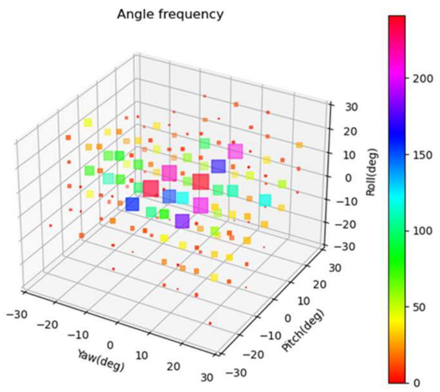

Fig. 5. Example 3D histogram showing the distribution of a face image subset, with relation to the pose angles. The angle space is separated into quadrants of 10 degrees. Larger squares signify higher frequencies of occurrence (left) and the color scale - correspondence of the number of face images (right).

978-1-6654-9878-4/22/$31.00 ©2022 IEEE

IV. Experiments and results

For our experiments, we have chosen the Pandora [13] face image dataset, which was recorded by a Kinect v2 sensor. The raw dataset is represented as sequences of 1920x1080 RGB pixel frames, for which the pose vectors are recorded.

The Pandora dataset contains 110 annotated video sequences using 10 male and 12 female actors. Each subject has been recorded five times. Different examples can be seen in Fig. 4. For the purpose of the experiments, classes of people, where the face is occluded in some of the images, have been discarded. In order to achieve a more balanced dataset, only faces with angles between -45 and +45 degrees were included. The angle space for each coordinate was split into 3 quadrants, resulting in 27 total quadrants. The filtered dataset consists of 18 different people, each of whom is represented by a subset of between 2500 and 3000 frames.

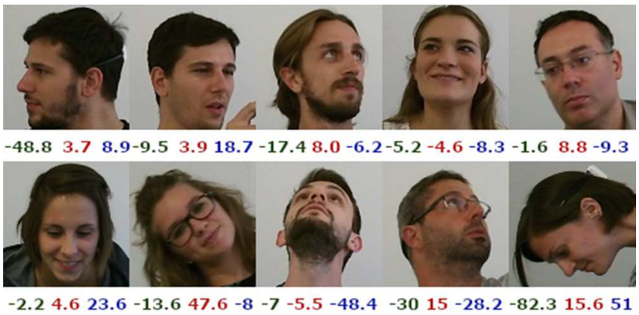

Fig. 6. Example images from Pandora dataset with their corresponding face angles in degrees (yaw - green, roll - red, pitch– blue). Note that the annotated angles in some of the frames are contradictory due to errors in pose estimation of edge cases.

A. Incremental learning

For this and the next experiment, the Pandora dataset was split into incremental steps, consisting of 2, 4, 8, 14 and 18 classes. 15% of all images were uniformly sampled from all classes and reserved for validation beforehand.
Only classes where the faces are fully visible were chosen (without hats or glasses).

The GAN model is trained for 90 epochs in every incremental step with a batch size of 64. The optimizer for both the generator and discriminator is Adam with a learning rate of 0.0002, $\beta_1 = 0.5$ and $\beta_2 = 0.9$. The detached supervised discriminator is trained with a learning rate of 0.001 or 10 epochs. Accuracies of old and new classes, as well as total accuracy of all classes, were measured at each step from the validation subset (Table 1).

Table 1. Accuracy results from Incremental training of Pandora dataset

Incremental step Metric	2	4	8	14	18
a_{old}	-	0,966	0,894	0,812	0,707
a_{new}	0,985	0,893	0,878	0,854	0,77
a_{total}	0,985	0,929	0,886	0,826	0,728

B. Incremental and active learning subsampling

The same experiment as before is performed however, this time for every incremental batch, specific samples are selected based on a predefined sampling budget of 1000 examples. Results are shown in Table. 2 and a comparison of both approaches is made in Fig. 7.

Table 2. Accuracy results from Incremental Active training of Pandora dataset for a budget of 1000 examples

Incremental step Metric	2	4	8	14	18
a_{old}	-	0,951	0,892	0,787	0,673
a_{new}	0,972	0,889	0,878	0,869	0,738
a_{total}	0,972	0,92	0,885	0,814	0,694

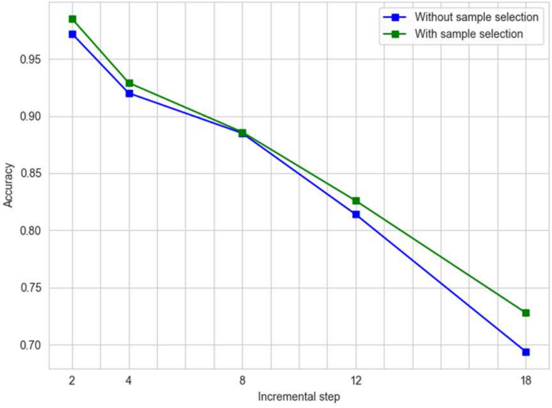

Fig. 7. Comparison of the total accuracies of the two training approaches

V. Conclusion

In this paper, we have proposed and tested a general method for real-time face annotation, which could be used in a closed door system with several capture devices. Results show that accuracies over consecutive steps decline but still show a viable increase with active learning. Even so, our experiments do not include as many classes and this could become problematic for an open system with no upper bound for incoming people. Furthermore, the time to process the images and fit them to the model needs to be taken into account. This is because frames arrive a lot faster than it is to train a model. Additional scenarios for those potential issues need to be implemented in the future.

ACKNOWLEDGMENT

This work was supported by the contract KP-06-H37/8 from 06.12.2019 for research project: “Inference algorithms for semantic knowledge extraction based on deep architectures for context-aware holographic communication” of the Bulgarian Research Fund of the Ministry of Education and Science, Bulgaria

REFERENCES

[1] G. Sang, F. He, R. Zhu, S. Xuan, "Learning toward practical head pose estimation," Opt. Eng. 56(8) 083104 (19 August 2017), DOI:10.1117/1.OE.56.8.083104

[2] S.Ye, R. Bohush, H. Chen, I. Zakharava, S. Ablameyko. “Person Tracking and Reidentification for Multicamera Indoor Video Surveillance Systems”. Pattern Recognition and Image Analysis, 30. 827-837. 10.1134/S1054661820040136.

[3] J. Heo, S. Marios – “Generic 3D face pose estimation using facial shapes”, 2011 International Joint Conference on Biometrics (IJCB), 1–8. doi:10.1109/ijcb.2011.6117472

[4] R. Ullah et. al. "A Real-Time Framework for Human Face Detection and Recognition in CCTV Images", Mathematical Problems in Engineering, vol. 2022, Article ID 3276704, 12 pages, 2022. https://doi.org/10.1155/2022/3276704

[5] P. Hristov, P. Nikolov, A. Manolova and O. Boumbarov, "Multi-view RGB-D System for Person Specific Activity Recognition in the context of holographic communication," 2020 XXIX International Scientific Conference Electronics (ET), 2020, pp. 1-4, doi: 10.1109/ET50336.2020.9238233.

[6] N. A. Syed, H. Liu, K. K. Sung. 1999. “Handling concept drifts in incremental learning with support vector machines.” In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '99). Association for Computing Machinery, New York, NY, USA, 317–321. https://doi.org/10.1145/312129.312267

[7] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, K.Alahari. End-to-End Incremental Learning. ECCV 2018 - European Conference on Computer Vision, Sep 2018, Munich, Germany. pp.241-257

[8] R. Kemker and C. Kanan, “FearNet: Brain-Inspired Model for Incremental Learning.” arXiv, Feb. 23, 2018.

[9] Y. Xiang, Y. Fu, P. Ji and H. Huang, "Incremental Learning Using Conditional Adversarial Networks," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6618-6627, doi: 10.1109/ICCV.2019.00672.

[10] Y .Kim, B. Shin (2022). In Defense of Core-set: A Density-aware Core-set Selection for Active Learning. arXiv preprint arXiv:2206.04838.

[11] Ye, S., Bohush, R.P., Chen, H. et al. Person Tracking and Reidentification for Multicamera Indoor Video Surveillance Systems. Pattern Recognit. Image Anal. 30, 827–837 (2020). https://doi.org/10.1134/S1054661820040136

[12] E. Belouadah, A. Popescu, U. Aggarwal, L. Saci, (2020, August). Active class incremental learning for imbalanced datasets. In European Conference on Computer Vision (pp. 146-162). Springer, Cham.

[13] G. Borghi, M. Venturelli, R. Vezzani and R. Cucchiara, "POSEidon: Face-from-Depth for Driver Pose Estimation," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5494-5503, doi: 10.1109/CVPR.2017.583.

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[15] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “Vggface2:A dataset for recognizing faces across pose and age,” in 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG2018). IEEE, 2018, pp. 67–74.

[16] Geng, X., Smith-Miles, K. (2009). Incremental Learning. In: Li, S.Z., Jain, A. (eds) Encyclopedia of Biometrics. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-73003-5_304

[17] T. Baltrusaitis, A. Zadeh, Y. C. Lim and L. -P. Morency, "OpenFace 2.0: Facial Behavior Analysis Toolkit," 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), 2018, pp. 59-66, doi: 10.1109/FG.2018.00019.

[18] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," in IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499-1503, Oct. 2016, doi: 10.1109/LSP.2016.2603342.

[19] Weinstein, B., Fine, S., & Hel-Or, Y. (2019). Selective sampling for accelerating training of deep neural networks. arXiv preprint arXiv:1911.06996.

[20] Seeland M, Mäder P. Multi-view classification with convolutional neural networks. PLoS One. 2021 Jan 12;16(1):e0245230. doi: 10.1371/journal.pone.0245230. Erratum in: PLoS One. 2021 Apr 8;16(4):e0250190. PMID: 33434208; PMCID: PMC7802953.