OAHEGA: From Research to Production of Mobile Expression Detection Application

Volodymyr Kovenko (urumipainblackreaper@gmail.com)
Vinnick'ij Nacional'nij Tehnicnij Universitet https://orcid.org/0000-0003-3825-1115
Vitalii Shevchuk
Vinnick'ij Nacional'nij Tehnicnij Universitet

Research Article

Keywords: facial, machine learning, emotion, detection

Posted Date: March 22nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1461348/v1

OAHEGA: FROM RESEARCH TO PRODUCTION OF MOBILE EXPRESSION DETECTION APPLICATION

Volodymyr Kovenko
urumipainblackreaper@gmail.com

Vitalii Shevchuk
1vitaliy.shevchuk1995@gmail.com

Abstract

Nowadays machine learning algorithms allow us to solve various difficult problems and optimize things a lot. One of such problems is detection of facial expressions. Detection of facial expressions can be used for market analysis, self-driving cars and in the entertainment industry. However, many challenges arise when trying to build a classifier for custom emotion, such as covariate shift and imbalanced number of instances per class. In this paper we present a new image dataset OAHEGA, that consists of six emotions, and conduct experiments on it. We also provide a comprehensive study of building a successful application on top of a model trained using the highlighted dataset.

Keywords: facial, machine learning, emotion, detection

1. Introduction

Recently, a type of machine learning algorithms called neural networks experienced a rapid increase in demand. Since the ImageNet competition and AlexNet [1] which was a winning architecture back in 2012, the computer vision field developed rapidly. Such a fast development is due to two main reasons: availability of big image datasets and construction of better algorithms. These improvements gave a possibility to solve a huge number of vision related tasks. One of such problems is understanding and detection of facial expressions. This task is crucial for many applications including market research, making self-driving cars safer, analysis of interviews and so on. The described task is not a new one, thus many solutions exist along with datasets, including FER [2] and AffectNet [3]. However, the task becomes more challenging if we consider detecting a new class of images because of challenges of data collection and covariate shift. In our work we describe the procedure of collecting a new dataset, containing six emotions: neutral, happy, angry, surprise, sad and ahegao, and show the experimental results on it. Big part of work is devoted to automation of data collection and processing using state-of-the-art object detection model YoloV3[4]. Building a real time emotion recognition system, embedded in mobile applications, raises new challenges of seeking for balance between models' accuracy and speed. Thus, we describe an overall pipeline consisting of face and body detection/localization model and an emotion recognition one. We make a choice of an architecture based on embedded devices limitations and experiment with possible optimizations using a knowledge distillation [5] technique along with post-training quantization [6,7]. Finally, a procedure of building a mobile application on top of a constructed pipeline and trained models is described.

2. Prior work

Paul Ekman is best known for his work with facial expressions. He theorized that not all expressions are the result of culture. Instead, they express universal emotions and are therefore biological. They include surprise, sadness, happiness, disgust, anger, and fear. One of the most famous emotion recognition datasets, FER, includes six universal emotions. Many researchers used this dataset as a main benchmark for facial expressions recognition. Pramerdorfer et al. [8] make an overview of six CNN based approaches to FER dataset and consider state-of-the-art CNN architectures like Inception[9], Res-Net[10] and VGG[11] with small changes. In their work it’s shown

that using an ensemble of recent deep CNN models, one can achieve a state-of-the-art performance on FER dataset without any need for additional image data and comprehensive data augmentation. Siqueira et al. [12] show that their ESR (Ensemble with Shared Representation) model copes with a problem of unbalanced label distribution and outperforms state-of-the-art deep neural networks on AffectNet and FER+ [13] datasets. One of the main problems of training every machine learning algorithm is the quality of the input data. As it's a very difficult job to annotate an emotion recognition dataset due to the uncertainties caused by ambiguous facial expressions and low-quality facial images, the quality of corresponding labels decreases. Wang et al. [14] propose Self Cure Network (SCN), that suppresses uncertainties efficiently and prevents deep networks from overfitting uncertain facial images. It's achieved by using three main building blocks: self-attention importance weighting, ranking regularization and relabeling. While their approach shows competitive results on CK+ [15], FER+ and AffectNet datasets, the results on the newly introduced WebEmotion dataset show that the highlighted approach is even capable of handling both synthetic and real-world uncertainties effectively. In the most recent work by Shi et al. [16], the researchers argue that the feature map is eroded after multi-layer convolution, which reduces the performance. As the solution to this problem, they introduce a novel architecture named Amend Representation Module (ARM). In order to cope with imbalanced data, a minimal random resampling (MRR) scheme is employed. Using both techniques together, researchers achieve current state-of-the-art performance on RAF-DB [17], FER2013 and AffectNet.

Though many new discoveries and developments were made in terms of model architectures to exceed the state-of-the-art performance on known benchmarks, the highlighted approaches were solely focused on the increase of accuracy. In our approach we are more focused on finding a balance between accuracy and speed, as needed for real-time performance on embedded devices. Similarly, to Hewitt et al. [18], we considered Mobile Net as a main model for emotion recognition, because of its highly optimized structure and competitive accuracy. As it was already exposed in the paper, one of the main problems in machine learning projects is the quality of the data. Thus we collect a new dataset with a custom emotion. Approaches of handling overfitting and imbalanced data along with model training and construction of the application are considered further in the paper.

3.Data gathering

The overall data gathering process was based on the constructed pipeline and functionality that was included in the final application. Recognition of emotions implies understanding of the face position, which can be challenging considering the end-to-end system. One of the workarounds is to add complementary information to the model as coordinates of face position or/and key points of the face. Another possible solution is to use a face detection system to first crop the face from the image and then use it as an input data to the emotion recognition system. As the main approach, the second one was chosen. Additionally, to face detection, it was decided to make people's detection. Based on these choices, the data gathering process was divided into two parts:

1) Gathering of the data for face and body detection.
2) Gathering an emotion detection dataset.

For face and people detection it was decided to use a neural network-based approach. Nevertheless, many pretrained models along with datasets for face detection (WIDER FACE [19], YouTube Faces [20], etc.) exist, the need for people detection made us make our own dataset with further model training. In order not to waste much time on the annotation process, the automotive labeling approach was used. Having a model trained to detect faces and the one trained for people detection, the data was labeled automatically. For both tasks, YoloV3 models were used, first one - trained on WIDER FACE and second one - trained on COCO (excluding all the classes, except "person"). Despite, there are other models which are built solely for solving the task of face detection and localization, such as MT-CNN introduced by Zhang et al[21], which uses multiple CNNs for the task of face detection (one for face detection, other one for bounding box regression and the last one for facial landmarks localization), it was decided to stick to the family of models we were familiar with. Logically, the data that contains facial information - also contains information about people's bodies. Thus, in terms of data, the WIDER FACE dataset was chosen (fig. 1).

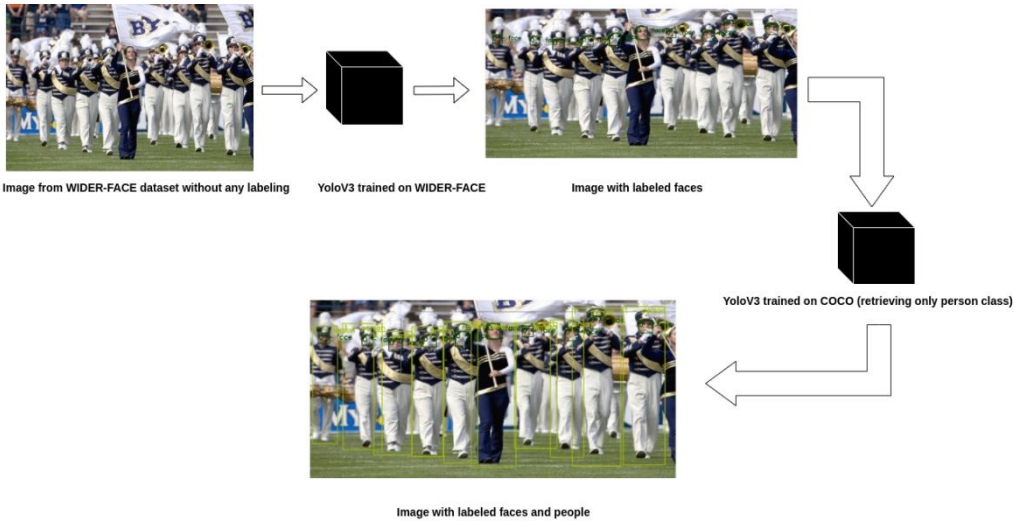

Figure 1 - The process of automotive data labeling for faces and people detection

Annotations were saved to TFRecord format for further training simplicity.

As it was already discussed in a paper, one of our main contributions is a new dataset [22] for emotion recognition. Two aspects made us collect a new dataset:

1) A problem of covariate shifts due to images with “ahegao” emotion.
2) An uncertainty in emotion labels and low quality of images.

In terms of data sources YouTube videos, photographs from Instagram and Facebook were used along with some images from AffectNet and IMDB datasets. During data collection each image was processed in the way that face was cropped from it using YoloV3 trained on WIDER FACE data. In result we have got a dataset with six emotions of high quality: neutral, happy, angry, surprise, sad and ahegao (fig. 2).

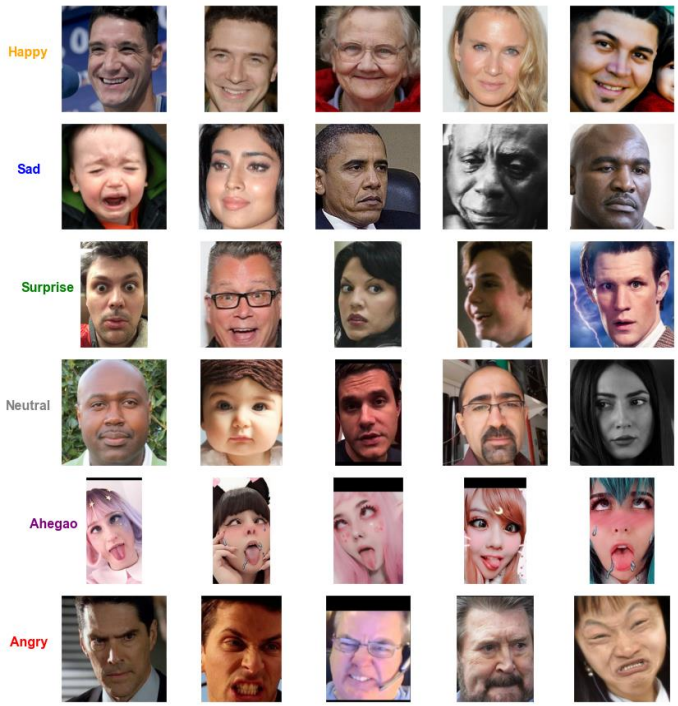

Figure 2 - The overview of OAHEGA dataset

It is important to note, that disgust and fear emotions weren’t included in the dataset, as of the fact, that the number of qualitative samples related to them was too low. The addition of other emotions both to the dataset and to the model is the matter of further work. One can also argue that as samples of a new emotion are very difficult to

collect and introduce the problem of covariate shift, possibly it would be better not to include it all . The thing is that this particular emotion is considered to be an “easter egg” and secret in the built application and is mandatory for the full entertainment of the end user.

The dataset can be thought of as the one that mixes two concepts, as rather than being biologically based, the ahegao emotion is inspired by internet fashion. Ahegao emotion itself denotes the state of great happiness and excitement and is a common expression in Japanese online communication. One of the ways used to visualize the data was using the extracted features from MobileNetV1[23] trained on ImageNet dataset and TensorFlow Embeddings Projector. OAHEGA dataset consists of 15744 images. The lowest number of samples is related to the “ahegao” emotion, as it was extremely difficult to find qualitative images of it (fig. 3).

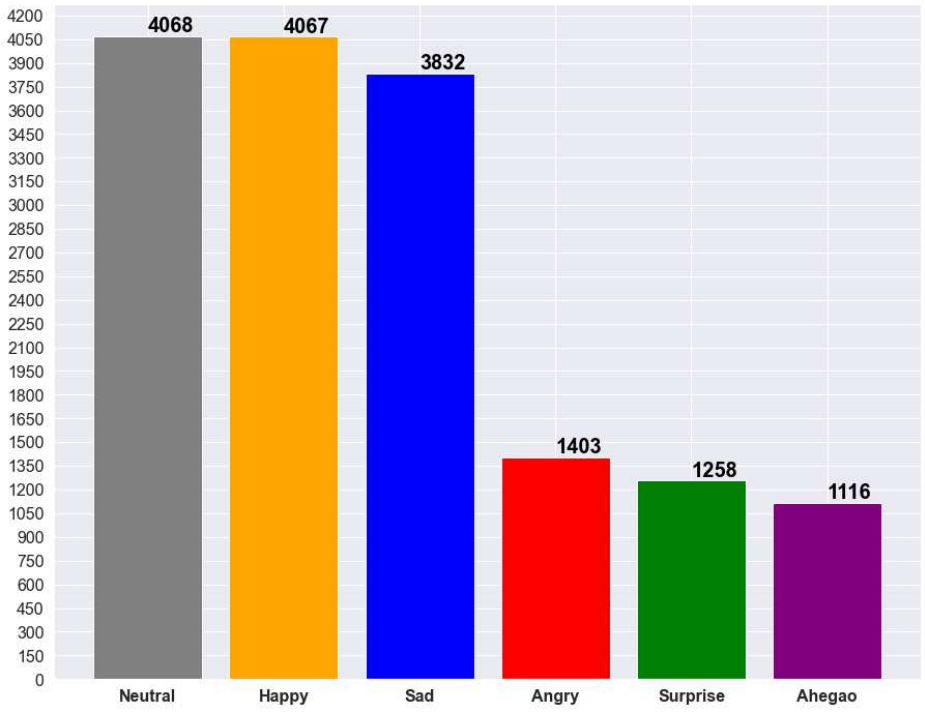

Figure 3 - Distribution of samples in each class

Though the presented dataset is of small size, it was enough to build a pretty accurate emotion classifier that satisfied the needs of application.

4. Models’ trainings

As our pipeline consisted of two different steps: detection of faces and people and emotion recognition, a separate model was trained for each step. For training of both models, the TensorFlow [24] package was used.

4.1 People and faces detection

Main requirements to the model for detection of faces and people was a high speed and accuracy, as it would be used on mobile devices and the performance of emotion recognition model depended much on it. Originally, the idea was to use the YoloV3 model, because of its brilliant performance. Unfortunately, it wasn’t optimized for speed, thus it would be too exhaustive in terms of resources to use it on embedded platforms. Another choice could be a Yolo Tiny, but as it was too difficult to port it to mobile devices, we didn’t choose it. What was really needed, is an algorithm that can provide a flexibility to impact speed-accuracy balance. One of such algorithms is SSD (Single Shot Detector) [25]. A main advantage of SSD is that it has a so-called backbone model, which is a sort of feature extractor. As a backbone model any good model trained for high quality image classification is used. After the features from the image are extracted, other layers are used to make multiscale detections. The possibility to choose a backbone model allows to tune the balance between accuracy and speed.

As a backbone model a MobileNetV1 was used. The overall MobileNetV1-SSD was trained on the collected dataset using TensorFlow detection model zoo.

4.2 Emotions recognition

As the emotion recognition step was a key one in our application, the corresponding model had to be as accurate as possible. Other requirements were a small size of model weights and a high speed of a prediction phase. According to the outlined needs, MobileNetV1 was used. MobileNet offers a possibility to control the number of parameters in a model through changing the alpha parameter. Alpha parameter is a width multiplier, which accepts values between 0 and 1. In our work, the alpha was set to 1. In order to speed up the training procedure, transfer learning [26] technique was used.

Transfer learning gives a possibility to transfer knowledge of the model from one domain to another one, by freezing part of weights and retraining upper layers of the network. As ImageNet didn't contain many useful features regarding facial expressions, only a part of lower layers was frozen, due to the fact that those layers were corresponding to the low-level features such as edges and shapes of objects. Number of layers to finetune was set as a hyper-parameter and was tuned. In order to increase the performance of the network, additional fully connected layers were added on top of the model and their number was defined as another hyper-parameter.

Training on the imbalanced classes can lead to the problem of overfitting, as the algorithm tends to pay too much attention to oversampled classes. Some of the possible solutions to the problem is oversampling of classes with small amounts of samples or downsampling of those which contain a bigger number of samples w.r.t other classes. On the one hand it was too difficult and cumbersome to continue data gathering, on the other hand we didn't want to miss important patterns and information by downsampling. Thus, in order to mitigate the problem, an imbalanced weight was utilized. Specifically, the imbalanced weights were calculated by formula 1.

$$w = \sum_{i=0}^{C} \frac{N_i}{N_{mean}},$$
$$w_i = \begin{cases} \frac{1}{w_i * \mu} \; if \; w_i < 1 \\ else \; 1 \end{cases}, \quad (1)$$

where C- unique classes, N - vector with number of samples for each class, w - vector of imbalances weights for each class, μ- a tunable parameter.

Overfitting can also happen due to the big complexity of the model. Dropout technique introduced by Srivastava et.al [27] is an efficient way of preventing overfitting in neural networks. During training of the algorithm, dropout randomly chooses a portion of neurons which is temporarily removed from the network and not updated. This procedure results in training a shallow network each time, making the overall network more generalizable. The dropout technique was utilized along with L2 weight decay.

Another powerful method of tackling the problem of overfitting and adding more variety to the data is image augmentation [28]. Image augmentation works by creating new samples by corruption, distortion and transformation of the existing ones. Nevertheless, there are many sophisticated approaches to image augmentation such as transferring the style of the other image on the original one using GANs [29] or learning the augmentation [30] we sticked to the classical approach of affine transformation. Portion of samples was zoomed out and horizontally flipped (fig. 4).

Figure 4 - Example of used augmentation

Such augmentation was chosen upon the fact that such distortions could happen in a real-world application.

Sometimes the optimization algorithm can be stuck in local minima, by converging to a suboptimal solution. This often happens because of a large learning rate. In order to fix the problem, the learning rate reduction technique is used. In our work, the learning rate was reduced with a factor of 0.5, when results on a validation set weren't improving for more than 10 epochs.

Training a machine learning algorithm implies choosing the right hyperparameters. Neural networks are often very sensible towards its setup, and changing such hyperparameters as learning rate, dropout rate or number of neurons can lead to varying results. Tuning a big amount of hyperparameters is a very tedious and time-consuming procedure. Thus, in order to find the best hyperparameters, a Bayesian hyperparameter tuning approach [31] was used. Specifically, the following hyperparameters were tuned: class weights parameter, number of epochs, batch size, dropout rate, initial learning rate, l2 regularization power, number of layers to finetune and number of dense layers. The hyperparameters tuning included 10 experiments and the bayesian algorithm was used for optimization w.r.t minimization of validation loss. As the optimizer for training the neural network, an Adam [32] was used. For each experiment, the following metrics were considered: accuracy, precision and recall. However, the precision was picked as the main metric (figure 5).

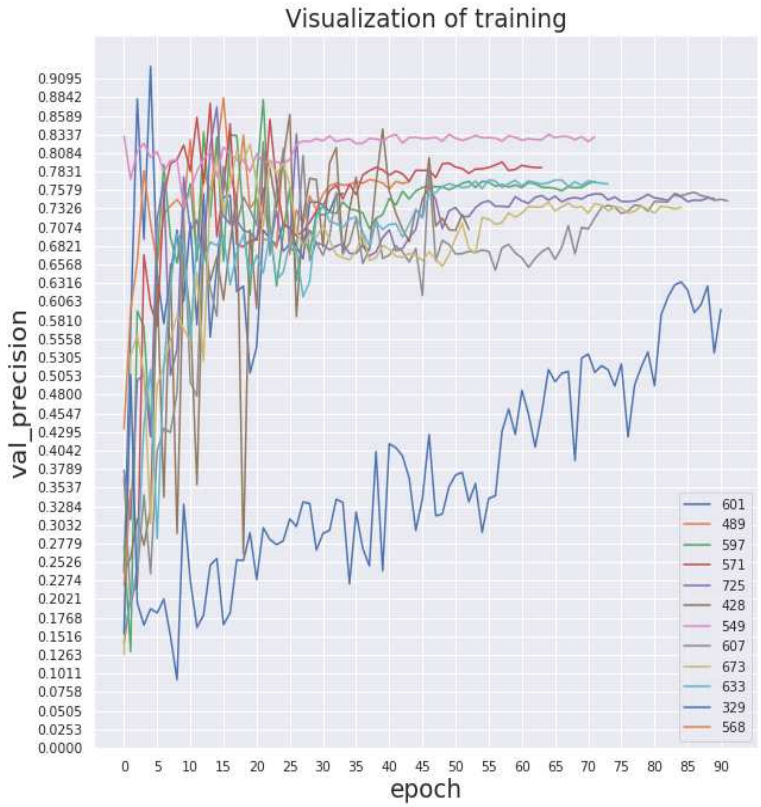

Figure 5 - Validation precision

As it can be seen from figure 5, the best result is achieved by experiment 549. The exact architecture of the model is presented on figure 6.

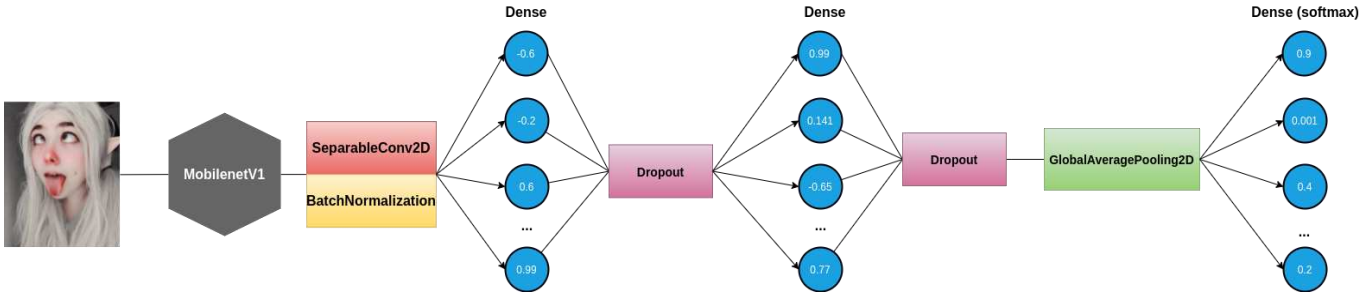

Figure 6 - best model architecture

From figure 6, the actual architecture of the model can seem to be strange because of a fully-connected layer right after the convolutional one. However, in that case, the result was a product of the weight matrix and slice of each tensor w.r.t last axis. To better understand the performance of each trained model, the precision per class was plotted (table 1).

Precision per class

	Angry	Ahegao	Happy	Neutral	Sad	Surprise	Average
Experiment							
549	0.79	**0.98**	0.93	**0.72**	**0.77**	**0.84**	**0.84**
571	0.73	0.97	0.93	0.65	0.73	0.80	0.80
633	**0.80**	**0.98**	**0.94**	0.67	0.62	0.75	0.79
489	0.74	0.95	0.93	0.65	0.69	0.81	0.79
597	0.76	0.96	0.93	0.65	0.65	0.75	0.78

Table 1 - precision per each class for 5 best experiments

From table 1, it's obvious that the hardest expressions to classify in all the experiments were Neutral and Sad. Probably, these two classes were misclassified between each other, as the emotions they express look pretty similar. It's also obvious why Ahegao and Happy gained the highest scores among all the models. Basically, these two emotions are the most visually distinct in the dataset. Nevertheless, the model trained during experiment 549 achieved best or similar precision for each class among other experiments. Thus, this model was picked for a production application.

5. Model's optimization

Deploying a machine learning model on cutting edge devices such as video cameras or mobile phones is a challenging task due to the limit of resources. Thus, many additional post-training procedures were created to shrink the size of the model and improve its inference time while losing from little to zero performance. Pruning [33] is a technique that works by removing neurons which have low sensitivity. It results in a sparse graph of computations and reduces the size of a model, while preserving its generalization ability. One minus of the highlighted approach is that, sometimes the model will have to be finetuned after pruning to restore the initial accuracy. Another approach is known as quantization. During quantization, the weights are compressed by lowering their precision. Quantization can be applied both during training and afterwards. The approach known is knowledge distillation presented by Hinton et al. is yet another technique for accelerating and compressing the model.

Unlike already discussed techniques, knowledge distillation is the process of transferring knowledge from a bigger model (called a teacher) to a smaller one (called a student). The student model is trained on the same dataset, the teacher one was trained, by minimizing the initial loss and the difference between outputs of its and teacher's logits. Minimizing the difference between outputs of logits, forces the student model to think as the teacher one, thus transferring the generalization ability of a bigger model. In our work, we focused on post-training quantization and knowledge distillation. While post-training quantization was applied both to object detection and

classification model, we experimented with knowledge distillation only w.r.t emotion classification one. For post-training quantization, a tflite package was used. Along with quantization, both models were transferred to tflite format, which is essentially a FlatBuffer, optimized for the inference on mobile devices. This optimization reduced the size of the object detection model from **21.7 MB** to **4.51 MB**, whereas for classification one - from **69 MB** to **5.5 MB**. For knowledge distillation, the original model was used as a teacher one and a smaller variant of MobileNet (with an alpha parameter equal to 0.25) as a student one. We firstly experimented with training a smaller model from scratch and using knowledge distillation. Each model was trained for 12 epochs and during distillation, the temperature for softmax was set to 2 (figure 7, 8).

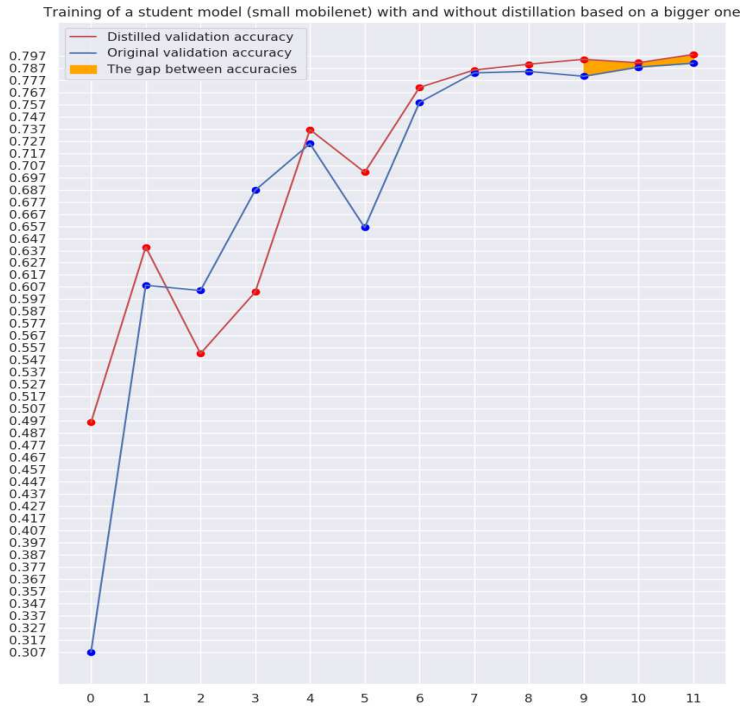

Figure 7 - comparison of accuracy of a model with and without distillation

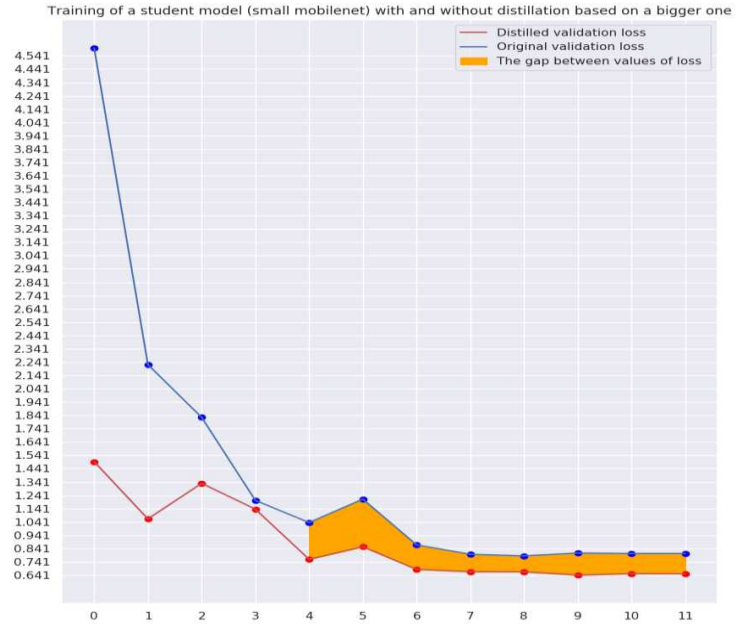

Figure 8 - comparison of loss of a model with and without distillation

As it can be seen, the distillation improved the accuracy of the student model compared to training from scratch. We then applied the same transformations as were made for the teacher model in order to prepare the distilled one for the production test. This procedure reduced the size of a distilled model from ~**4 MB** to **1.3 MB**. Second experiment was related to the classification speed on two devices: Samsung S10e and Xiaomi Mi 9T (figure 9).

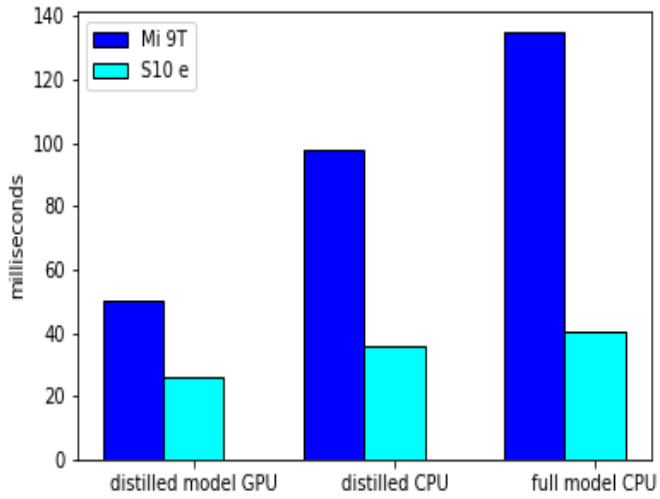

Figure 9 - performance comparison on two devices w.r.t teacher model and distilled model

Nevertheless, the distilled model has shown great results in terms of speed and size, the performance on real world data was lower than we hoped. The cause of it could be in the fact that the specificity of training a teacher model, mainly augmentation, helped it to deal with production data, whereas distillation procedure wasn't powerful enough to fully transfer its generalization ability to a student one. However, many new approaches of distilling knowledge in neural networks were proposed, the discussion of them is out of scope of our paper.

6. Actual pipeline

Having trained the models and picked best for production application, the development of complementary logic was started. As the main language for application development, Java was chosen. The application was integrated with the TensorFlow-lite module for efficient usage of trained models. On top of it, additional logic was added. It was found that making emotion predictions on one frame wasn't efficient nor accurate enough. Thus, the logic for averaged predictions by frames was added (figure 10).

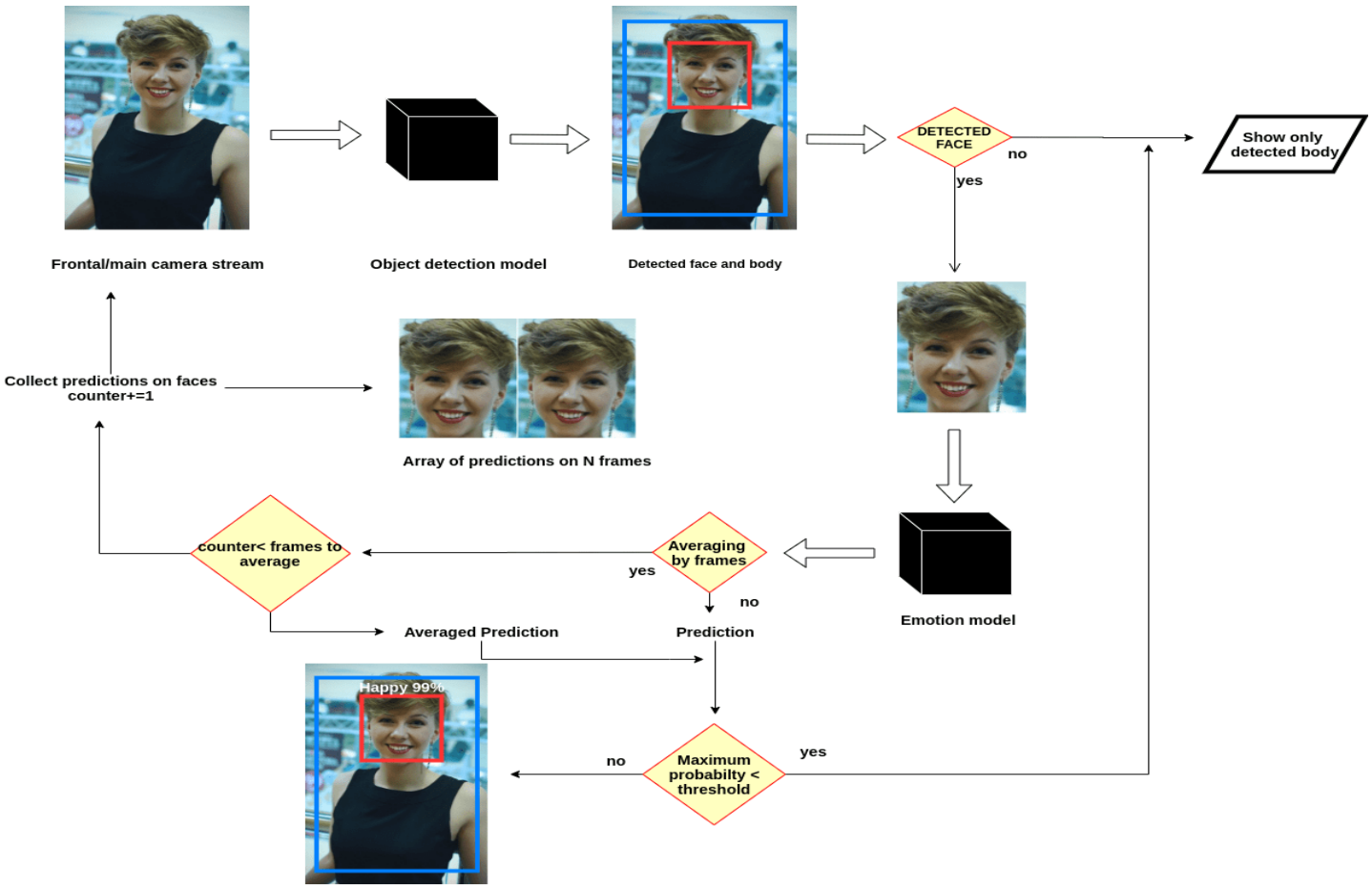

Figure 10 - scheme of application workflow

As it can be seen from figure 10, the application is working with a video stream from the main or frontal camera. Each frame goes through the object-detection model to locate face and body on the image. If the face is detected, it is cropped from the frame and used as the input to emotion detection model. If the averaging is enabled, the prediction is made for N frames and results are averaged by probabilities. The last piece of logic is related to showing a prediction only if its probability satisfies the minimal threshold. The design of the application allows configuring such parameters as number of frames to average and minimal probability threshold in the settings screen (figure 11).

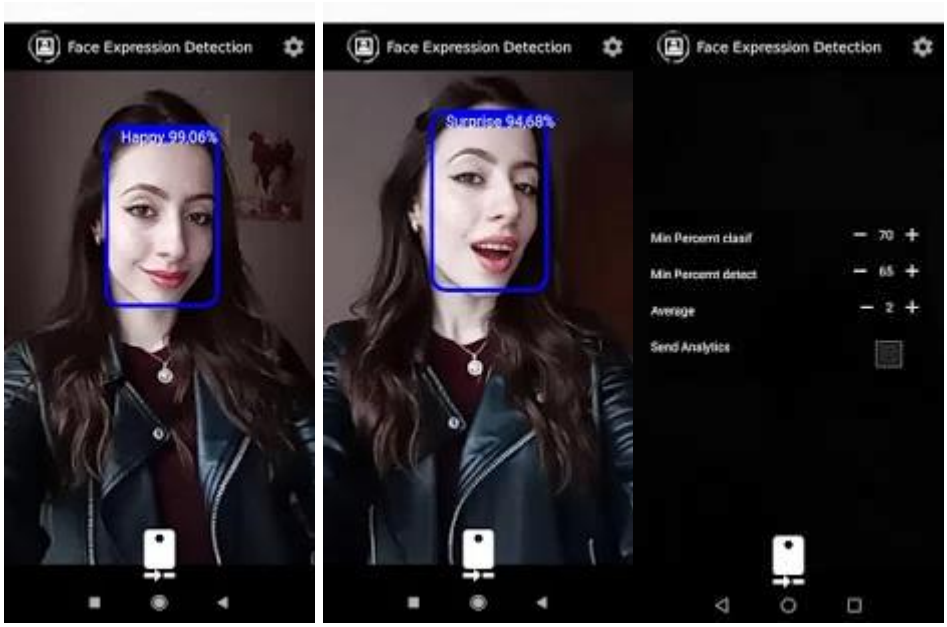

Figure 11 - example of application design

The application was released in Google Play [34].

Conclusion

In this work, the process of developing a machine learning application for expression detection was described. The discussion touched on data gathering, models' training and optimization and final application assembling. We presented a new unique medium-size dataset which can be used to solve the problem of emotion classification, described the process of automated data labeling and showed model training w.r.t gathered data. Nevertheless, the application was made only for entertainment and educational goals, the theme of emotion detection has a big potential. For instance, integration with embedded analytics due to emotions can grant a personalized and useful experience to the end user. Other possible improvements of application are related to new algorithms. Efficient Net presented by Tan et al [35] is a nice candidate to substitute the MobileNetV1, as it is superior both in terms of accuracy and speed to the last one. More research due to distillation can help to even more reduce model size and optimize it for inference. For example, the distillation process based on adversarial distillation or multimodal one can be tried. Finally, retraining models based on user data can allow models to become more generalizable and possibly nullify the problem of data drift.

Acknowledgments

We would like to thank Yevheniy Sakun for the useful help with application development.

Funding

No grants were associated with this work.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou & K. Q. Weinberger (ed.), Advances in Neural Information Processing Systems 25 (pp. 1097--1105) . Curran Associates, Inc. .

[2] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio. Challenges in representation learning: A report on three machine learning contests. Neural Networks, 64:59--63, 2015. Special Issue on "Deep Learning of Representations"
[3] Ali Mollahosseini, Behzad Hasani, and Mohammad H. Mahoor, “AffectNet: A New Database for Facial Expression, Valence, and Arousal Computation in the Wild”, IEEE Transactions on Affective Computing, 2017.
[4] Redmon, Joseph & Farhadi, Ali. (2018). YOLOv3: An Incremental Improvement. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1804.02767.pdf – Title from the screen.
[5] Hinton, G., Vinyals, O. & Dean, J. (2015). Distilling the knowledge in a neural network. Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1503.02531v1.pdf – Title from the screen.
[6] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1806.08342.pdf – Title from the screen.
[7] Jacob et al. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1712.05877.pdf – Title from the screen.
[8] Christopher Pramerdorfer, Martin Kampel. Facial Expression Recognition using Convolutional Neural Networks: State of the Art [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1612.02903.pdf – Title from the screen.
[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1512.00567.pdf – Title from the screen.
[10] Kaiming He et. al. Deep Residual Learning for Image Recognition. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1512.03385.pdf– Title from the screen.
[11] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015, pp. 730-734, doi: 10.1109/ACPR.2015.7486599.
[12] Henrique Siqueira, Sven Magg and Stefan Wermter. Efficient Facial Feature Learning with Wide Ensemble-based Convolutional Neural Networks. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1612.02903.pdf – Title from the screen.
[13] Emad Barsoum, Cha Zhang, Cristian Canton Ferrer and Zhengyou Zhang. Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1608.01041.pdf – Title from the screen.
[14] Kai Wang, Xiaojiang Peng, Jianfei Yang, Shijian Lu, and Yu Qiao. Suppressing Uncertainties for Large-Scale Facial Expression Recognition. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/2002.10392.pdf – Title from the screen.
[15] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2010, pp. 94-101, doi: 10.1109/CVPRW.2010.5543262.
[15] Jiawei Shi, Songhao Zhu , Zhiwei Liang. Learning to Amend Facial Expression Representation via De-albino and Affinity. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/2103.10189.pdf – Title from the screen.
[16] Jiawei Shi, Songhao Zhu , Zhiwei Liang. Learning to Amend Facial Expression Representation via De-albino and Affinity. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/2103.10189.pdf – Title from the screen.
[17] Charlie Hewitt, Hatice Gunes. CNN-based Facial Affect Analysis on Mobile Devices. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1807.08775.pdf – Title from the screen.
[18] Shuo Yang, Ping Luo, Chen Change Loy, Xiaoou Tang. WIDER FACE: A Face Detection Benchmark. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1511.06523.pdf – Title from the screen.

[19] Lior Wolf, Tal Hassner and Itay Maoz. Face Recognition in Unconstrained Videos with Matched Background Similarity. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011.
[20] Kovenko, Volodymyr; Shevchuk, Vitalii (2021), “OAHEGA : EMOTION RECOGNITION DATASET”, Mendeley Data, V3, doi: 10.17632/5ck5zz6f2c.3
[21] J. Xiang and G. Zhu, "Joint Face Detection and Facial Expression Recognition with MTCNN," 2017 4th International Conference on Information Science and Control Engineering (ICISCE), 2017, pp. 424-427, doi: 10.1109/ICISCE.2017.95.
[22] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv, abs/1704.04861.
[23] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
[24] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., & Berg, A. (2016). SSD: Single Shot MultiBox Detector. ECCV.
[25] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1411.1792.pdf – Title from the screen.
[26] Srivastava, Nitish & Hinton, Geoffrey & Krizhevsky, Alex & Sutskever, Ilya & Salakhutdinov, Ruslan. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 15. 1929-1958.
[27] Shorten, Connor & Khoshgoftaar, Taghi. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data. 6. 10.1186/s40537-019-0197-0.
[28] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. CoRR, abs/1703.10593, 2017
[29] Jason Wang, Luis Perez. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1712.04621.pdf – Title from the screen.
[30] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical Bayesian optimization of machine learning algorithms. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'12). Curran Associates Inc., Red Hook, NY, USA, 2951–2959.
[31] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.
[32] Xin Dong, Shangyu Chen, and Sinno Jialin Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1705.07565.pdf – Title from the screen.
[33] Oahega - Emotion detector. [Electronic resource] – Electronic data. – Mode of access: https://play.google.com/store/apps/details?id=org.oahega.com – Title from the screen.
[34] Mingxing Tan, Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. [Electronic resource] – Electronic data. – Mode of access: https://arxiv.org/pdf/1905.11946.pdf – Title from the screen.