Using the Temporal Data and Three-dimensional Convolutions for Sign Language Alphabet Recognition

Serhii Kondratiuk[a,b], Iurii Krak[a,b], Vladislav Kuznetsov[a] and Anatoliy Kulias[a]

[a] *Glushkov Cybernetics Institute, Kyiv, 40, Glushkov ave., 03187, Ukraine*
[b] *Taras Shevchenko National University of Kyiv, Kyiv, 64/13, Volodymyrska str., 01601, Ukraine*

Abstract

Research is the development of the communication technology for detection of gestures of Ukraine sign alphabet. Basis of neural network with mobilenet architecture was improved with a new deep learning architecture – mobilenetv3. Another improvement lies in the field of the dataset – additional portion of train dataset was collected, and test dataset became more diverse, also due to improved data augmentation techniques. Deep learning model was improved with three dimensional convolution and data processing technique in a form of temporal frames. A lot of experiments to have a consistent improvement in model performance with better data augmentation, novel mobilenetv3 architecture as a basis and spatio-temporal frame are demonstreted the effectiveness proposed approach.

Keywords [1]
gesture detection, 3d convolutions, mobilenetv3, temporal data, data augmentation, sign language

1. Introduction

The research is based on previous work in the field of gesture detection, specifically Ukrainian sign language [1]-[3]. The previous work introduced a new communication technology, both for studying and testing of Ukrainian sign language, with possibility to scale with other languages. One of the most prominent features was cross-platform of the technology [4]-[6].

Sign language is a frequently used method of communication among people with special communication needs. Those with hearing disabilities may utilize supplemental software to connect with society and within their own group. The dactyl alphabet should be learned utilizing gesture recognition technology.

With development of deep learning network and hardware which is a sufficient fit to train such a network, the gesture detection was implemented using a convolutional neural network using novel architecture mobilenet, which allowed to deploy the technology on mobile phones and for it to be a truly cross-platform.

Further improvement to the approach is present in the research, specifically the three-dimensional convolutions with the use of spatio-temporal data frame, which in combination with the newer mobilenetv3 architecture allows to achieve higher quality of sign recognition, due to ability to detect not only static but also dynamic features. Configuration and optimization of the approach are also part of the research.

[1]CMIS-2022: The Fifth International Workshop on Computer Modeling and Intelligent Systems, Zaporizhzhia, Ukraine, May 12, 2022
EMAIL: sergey.kondrat1990@gmail.com (S.Kondratiuk); yuri.krak@gmail.com (I.Krak); kuznetsow.wlad@gmail.com (V.Kuznetsov); anatoly016@gmail.com (A.Kulias)
ORCID: 0000-0002-5048-2576(S.Kondratiuk); 0000-0002-8043-0785(I. Krak); 0000-0002-1068-769X(V. Kuznetsov); 0000-0003-3715-1454 (A.Kulias)

2. Existing research

Sign gesture detection is a complex task, due to human hand being a highly dynamic object, able to represent signs with different peculiarities. Also, signs can be observed from different angles in different scales and in both axes. Another issue is lighting condition and physical parameters of the hand itself and surrounding environment.

Various algorithms based on conventional computer vision with hand-crafted features, such as the histogram of oriented gradients [7], bag-of-features [8], hyperplanes separation [9]. Unfortunately, due to mentioned before, peculiarities of the hand and it’s high-dynamic nature, such approaches are not robust enough and suffer a lot from changes in environment, background, or quality of the input images. Also, they are not scalable with the enlargement of image database of hand gestures samples.

Current state-of-the-art hand gesture recognition architectures [10], [11], [12] are based on convolutional neural networks, because they can overcome the issues with the environment and become robust to changes in background and surrounding, scale of the image and hand within it, and quality of the image itself. Another prominent feature is the ability of the convolutional neural network to improve as the training dataset grows and becomes more diverse.

3D convolutions allowed the same models to benefit from sequence of pictures, e.g., videos with activities. They show high performance with large amounts of data, even in the form of pre-recorded videos. (AlexNet [13], Sports-1M [14], Kinetics [15], Jester [16]). No overfitting happens with such huge and diverse datasets.

In order train and deploy deep learning models on low performance devices, such as smartphones, a subset of lightweight architectures with typically smaller amount of parameters and more effective structure were developed (SqueezeNet[17], MobileNet[18], MobileNetV2 [19], ShuffleNet [20] and ShuffleNetV2 [21], MobileNetV3 [22]). [23] presents Sequential Pattern Mining for tree topologies based recognition in their research.

3. Proposed approach

The proposed approach lies in two domains:

- Improving techniques for data augmentation
- Presenting, improving, and configuring approach of spatio-temporal data.

Such development allows to make the trained model become more robust to changes in input data, without the need to collect more of the datasets manually. On the other hand, it allows to improve the approach of detecting a sign on an image sequence (or video), and gestures are mostly a highly dynamical object, whilst transitions from one gesture to another could be highly diverse, and a model would benefit a lot from training on such transitions. All dataset processing and model training was performed using cross-platform framework [24].

Another improvement is in using a more novel, advanced, and optimized architecture as a basis – mobilenetv3.

3D convolution is used to improved detection with sequence of images (video). Such developments [25]-[27] show significant improvement of architectures with such enhancements in tasks with dynamic activities. Combination of lightweight architecture with such 3d convolutions allows to improve performance with dynamic sign detection on videos and maintain a possibility to deploy cross-platform, even on a low-performance device such a smartphone.

Spatio-temporal detectors can build spatial descriptors that include both spatial and temporal information. Both single images and sequences of images can be utilized as input for the model in the research.

It is possible to train the model to be more resistant to change in such a dynamic object as hand by analyzing multiple adjacent images in the sequence at once. This allows the network to be taught to consider the temporal aspect, i.e., the dynamics of change in movements in multiple input images. If an image contains artifacts, bad lighting, is fuzzy, or is obstructed in some manner, spatio-temporal approach can be used to smooth things out by utilizing surrounding frames in sequence.

4. Spatio-temporal frame

The idea of the of the spatio-temporal frame lies in concepts: to collect all the spacious information from the image, needed for the convolutional network to detect gestures, and second, to merge multiple frames information, to collect temporal component of the data, meaning dynamic changes from image to image. Only three-dimensional convolutions would be able to process such input and thus be able to detect patterns in temporal component over a sequence of input images.

A single image sequence could be treated as a single training sample for a convolutional neural network with 3d convolutions. However, is it unclear in such case, which would be the required length of the video. Moreover, it would limit somehow the format of the input data, requiring it to be of a specific predefined length or duration.

As a solution to such issues, in the research, instead of setting a requirement to input video, a concept of spatio-temporal frame was introduced.

It is similar to a floating window concept, which is widely used in object detection, when the image is passed in a predefined order with a window of a size significantly smaller than size of the image. Similarly, the spatio-temporal frame has a predefined size (for instance, 8 frame) and goes through the video of arbitrary length in a chronological order. Another major concept is the overlapping of the spatio-temporal frame, in other words, how many last frames in previous frame and first frames in next frame are the same. Having this parameter set too high, the result will be in excessive amount of data which will be generated by the preprocessing of training dataset and, finally, the neural network will train redundantly huge amount of data, which is mostly the same, also taking longer amount of time to train until convergence.

Thus, we have two hyper-parameters, define at the step of preprocessing of the dataset (splitting videos into spatio-temporal frames of fixed length) – size of the frame and number of overlapped frames. These parameters affect architecture, speed, and quality of the trained model, so they such be tuned carefully, and such tuning process was a part of the research, and optimal parameters were defined based on model performance. These hyperparameters also affect model architecture (size of layers).

Therefore, single spatio-temporal frame can be presented as:

$$D = \{d_{i-k},...,d_i,...,d_{i+k}\},\ i = \overline{1, n-k}, \tag{1}$$

where k is the number of previous and subsequent frames from the current, from which a sequence of images is formed (Fig. 1).

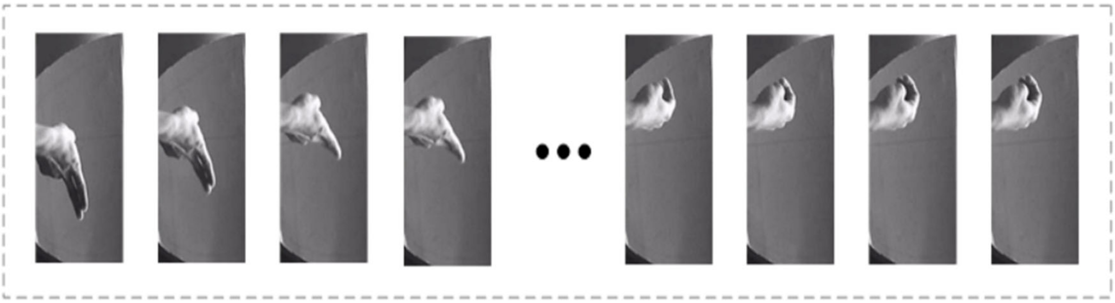


Figure 1: Two subsequences created from a single video stream

Figure 2 shows the schema in which one video is divided into spatio-temporal frames with optimal parameters.

Also, during split into train set and test set, a distribution of the data should be maintained in terms of size, quality, focal length, lighting, background, artifacts, blur and etc.

A uniform data processing approach was designed to convert them to a generic form for further computations inside the specified recognition model, both at the training and recognition stages.

As a result, from one video with a sign it is possible to obtain multiple sub sequences.

The process of tuning such hyper-parameters requires a pipeline, which consists of

- dataset preprocessing

- model architecture selection
- model training
- testing on a predefined test set (for the testing to be in fair conditions)

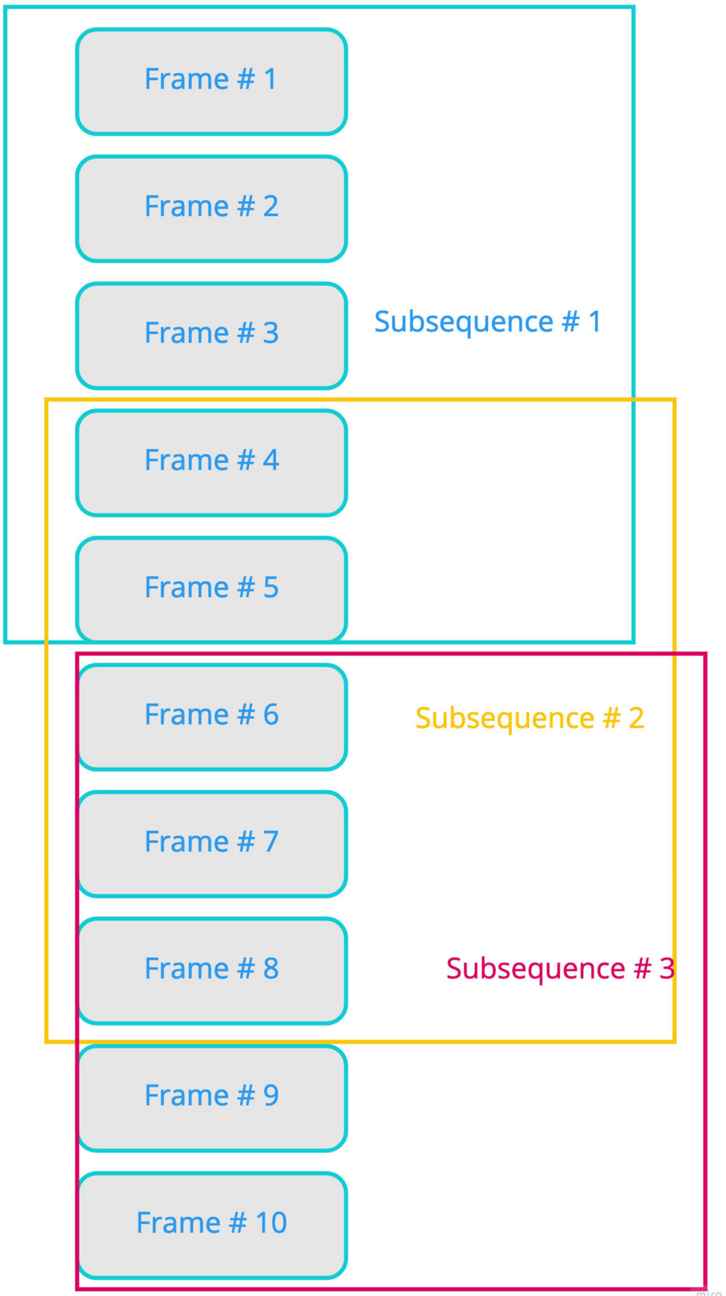


Figure 2: Schema in which one video is divided into spatio-temporal frames with optimal parameters.

There are three steps of dataset preprocessing:

- denoising
- resizing
- normalization

The schema of a pipeline which was used in the research is show at Figure 3.

The new MobileNetV3 architecture (Fig. 4) is an improvement of it's previous versions – mobilenet and mobilenetv2. Previous work used mobilenetv2 as a basis for convolutional neural net architecture, which was afterwards enhanced with 3d convolutions. As a part of previous work development, novel mobilenetv3 architecture is used in the research, also enhanced with 3d convolutions. Newest mobilenet reincarnation also has two versions – large and small, oriented on high and low hardware resource respectively. Also, a redesign of expensive layers took place, which allowed to further improve performance speed on all platforms.

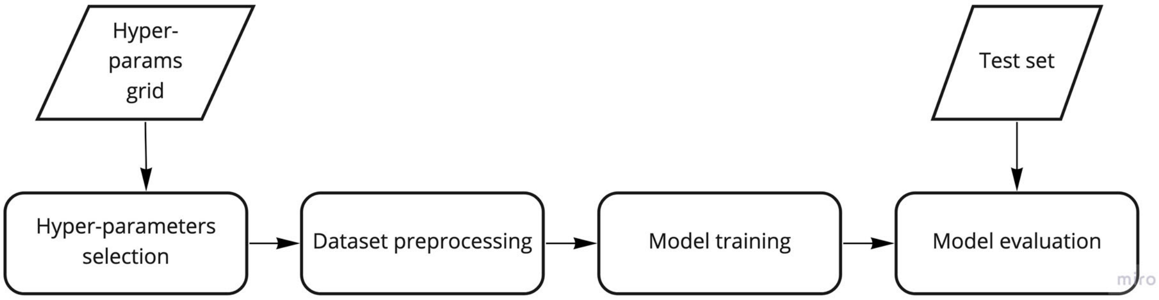
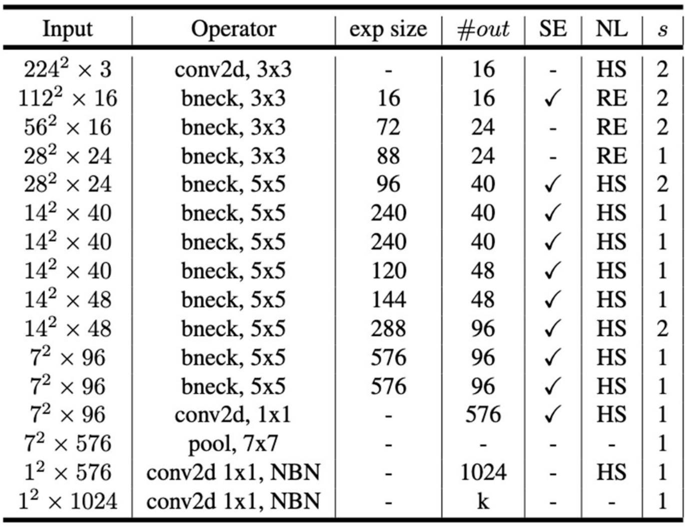


Figure 3: Schema of a pipeline for tuning spatio-temporal frame parameters.



Input	Operator	exp size	$\#out$	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

Figure 4: Architecture of MobileNetv3-small

In MobileNetv3 there have been two strategies to improve the network:

- Platform-aware NAS for block-wise search
- NetAdapt for layer-wise search

Introducing a dynamic and temporal component into the technology presented new challenges into the results interpretation. One of the issues which appeared in the process became instability of the prediction on dynamic data. In other words, a sequence of frames produces a sequence of predictions, but there can be wrong predictions, or the prediction could start frantically change from frame to frame.

As a part of improvement of the technology for such cases, an approach for stabilization of predictions was presented and implemented. The are two components of such solution: smoothing of the prediction probabilities and accumulation of probabilities from previous spatio-temporal frames. Also additional anomaly-detection approach is used to neglect predictions within a frame which are highly different from the context.

With such approach, first prediction of the first frame is considered as baseline. Further predictions on next frames start accumulating with the previous prediction, an in case if it’s highly different from a history of multiple predictions before – such an anomaly is neglected.

Predictions from earlier subsequences are used to build up the model, which then uses that information to update the current recognition result only when the total number of predictions surpasses a certain threshold.

$$\sum_{t=t-n}^{t+n}\sum_{i=t-k}^{t+k} p_i > threshold\ , \tag{2}$$

where: p_i - the probability of a gesture in the frame; i - frame number on the current subsequence; t - number of the current subsequence; k - the size of the subsequence in both directions; n - number of accumulated subsequences.

5. Dataset of sign images and their augmentations

In previous work a dataset of 50000 images was collected, with different sign, corresponding to Ukrainian sigh alphabet (Fig. 5). During creation of such a dataset, it was aimed at to make it diverse in terms of lighting conditions (20 % of data in bad of light conditions, 30 % in mediocre light conditions and 50 % in good quality lighting). Almost 10% of data was is poor quality, with noise and very blurry.

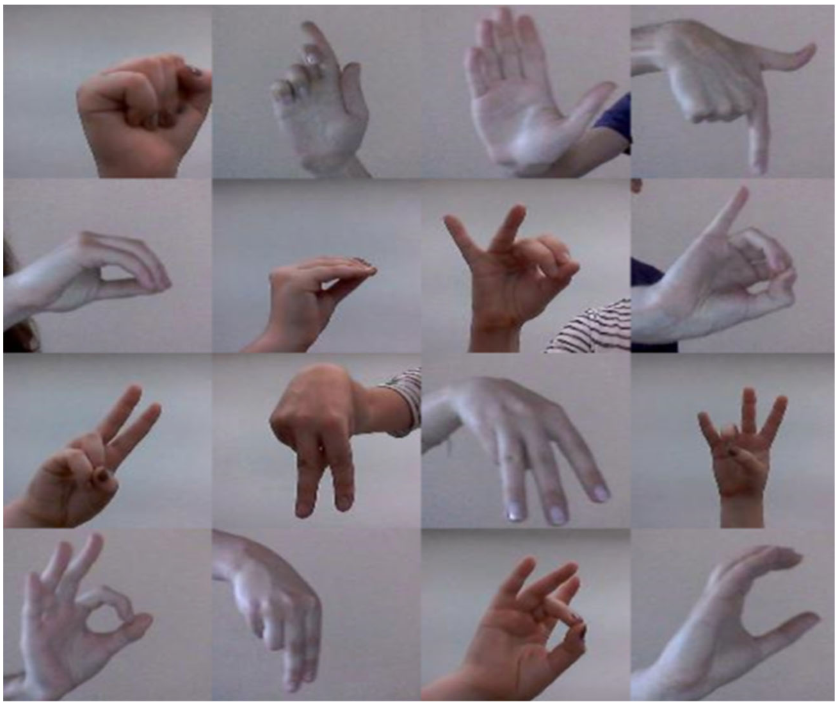


Figure 5: Dataset subsample

In order to increase the dataset size, a list of data augmentation techniques from previous work was enlarged, thus next data augmentation techniques were used:

- rotation
- random cropping
- flipping
- brightness adjustment
- noise addition
- salt and pepper (replaces pixels in images with salt/pepper noise (white/black-ish color))
- coarse dropout - sets rectangular areas within images to zero.
- gamma contrast - adjust image contrast by scaling pixel values to
- affine
- blur
- emboss

As a result, the train dataset was increased in 5 times and a final amount of 250,000 pictures was generated. 20% of the data was used as testing subset for pipeline for spatio-temporal hyperparameter tuning. Some additional data augmentation techniques were used exclusively to the test dataset (of size 50,000 pictures). This was done in order to verify that the technology will stay robust even in unseen before conditions and environment.

Figure 6 shows examples of original images and their augmented counterparts.

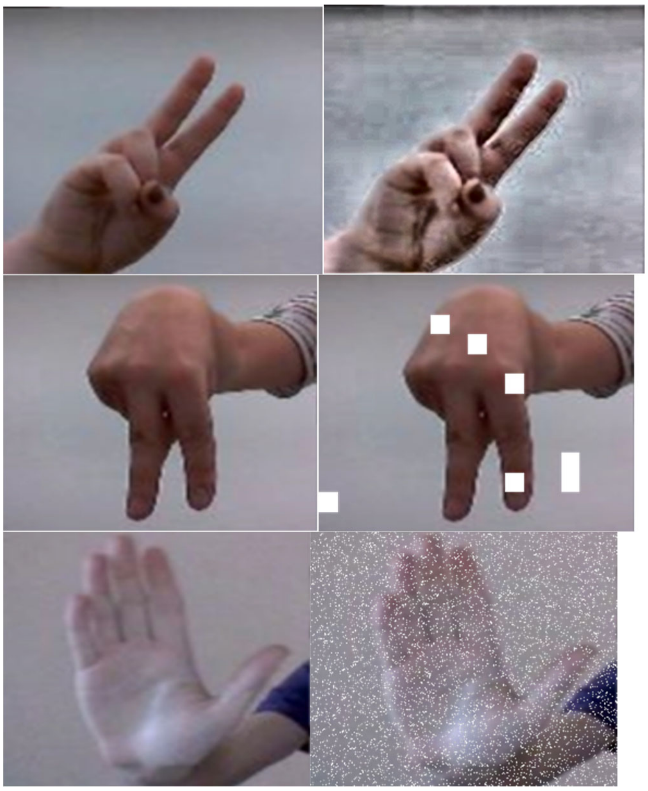


Figure 6: Original images (left) and augmented images (right). Topmost – heavy augmentation (multiple at once), middle – dropout, bottom – salt and pepper.

Figure 7 shows percentage of images with different light conditions in the train and test split datasets, divided into such conditions: bad, mediocre and good.

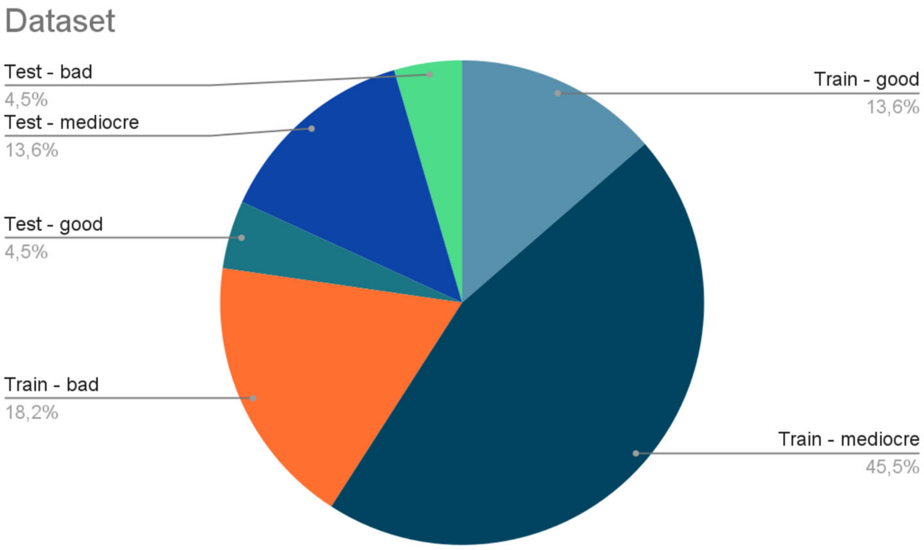


Figure 7: Distribution of light quality on the train and test datasets.

It is also important not to overfit the model with artificial patterns present in the dataset, thus it must maintain statistically significant variety. Also, there should not be major shifts in train dataset comparing to test dataset. All these conditions were met in the train dataset conducted and augmented as a part of research.

6. Experiments

During dataset augmentation of test split with techniques, which were not used in the train split, performance of the model dropped, naturally, due to higher complexity of the testing data. However, still it showed performance comparable with state-of-the-art approaches (for instance, squeeze-nets). Figure 8 shows performance increase in using more novel mobilenetv3 model as a basis, and also increase in performance with three dimensional convolutions. All measurements show macro-averaged f1-score.

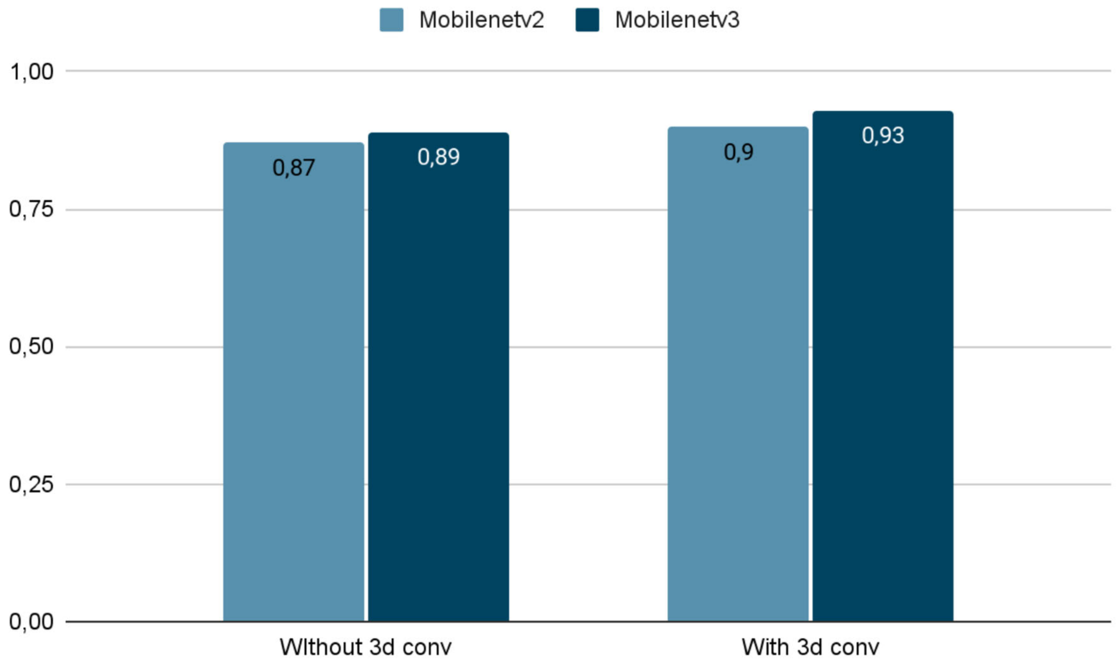


Figure 8: Comparison of models

During models training, hyperparameters of spatio-temporal frame were tuned first. After those parameters fixed, multiple model architectures varieties were considered, with mobilenetv3 as a basis for all of them. Mobilenetv3-small was preferred over mobilenetv3-large due to cross-platform considerations, with ability of such model to perform with relatively same gesture recognition performance but on a lower performance hardware, such as smartphones. Experimental results proved small version to be sufficient for gesture model. At least five different modifications of mobilenetv3-small architecture were tuned and evaluated during the experiments. It is important to analyze confusion matrix of model prediction. This also helped to select the best approach and best configuration. All of these measures allowed to design a balanced neural network that was both small and effective on the test data set.

Hyperparameters form a grid, which contains a configuration for model architecture, training hyperparameters and spatio-temporal hyperparameters.

Example grid:

$$\left\{\begin{array}{c}\text{learning\_rate: [0.001, 0.0001],}\\ \text{batch\_size: [8,16,32],}\\ \text{layers\_config: [config1, config2,config3],}\\ overlapping\ frames\text{: 2,}\\ frame\_size\text{: 5,}\\ decay\text{: 1}\end{array}\right. \tag{3}$$

Each model train was performed with common techniques for fighting overfitting. Model's prediction time is sufficient for real-time (24 fps) performance using Nvidia K80 GPU.

7. Conclusions

As a result of research, a technology for gesture communication was developed and improved within multiple domains. A novel architecture of mobilinetv3 was used as a lightweight neural network model, which allows both to get a high level of gesture recognition performance with ability to run on a low-performance hardware, such as smartphones. To overcome recognition issues with such a high dynamic object as hand and gesture animation, a three-dimensional convolution technique was used to enhance the neural network architecture. As a solution to use as input videos of different length, a spatio-temporal frame concept was introduced and implemented as a part of research. Having its own hyperparameters, such as number of overlapping frames and size of the frame, it needed configuration. As a part of research, best hyperparameters were tuned, both for spatio-temporal frame and for the neural network itself. A special pipeline was built for tuning hyperparameters for spatio-temporal frame. Additional data augmentation techniques were used to increase the train dataset, also some techniques were used only for test split of dataset, to prove robustness of the model to unseen conditions.

It was proved with experiments to have a consistent improvement in model performance with better data augmentation, novel mobilenetv3 architecture as a basis and spatio-temporal frame approach, which resulted in 0.93 macro-score f1.

8. References

[1] Kondratiuk S. Gesture recognition using cross platform software and convolutional neural networks // Artificial Intelligence. – b.83-84 – 2019 – p.94-100.

[2] S. Kondratiuk, I. Krak, A. Kylias, V. Kasianiuk. Fingerspelling Alphabet Recognition using Cnns with 3d Convolutions for Cross Platform Applications. Advances in Intelligent Systems and Computing. Vol. 1246 AISC. 2021, pp.585-596. doi:10.1007/978-3-030-54215-3_37.

[3] Yu.V.Krak, Yu.V. Barchukova, B.A. Trotsenko. Human hand motion parametrization for dactylemes modeling, Journal of Automation and Information Sciences, 43(12) (2011):1-11. doi:10.1615/JAutomatInfScien.v43.i12.10

[4] The Linux Information Project, Cross-platform Definition. www.linfo.org

[5] Y.V. Krak, A.A. Golik, V. S. Kasianiuk. Recognition of dactylemes of Ukrainian sign language based on the geometric characteristics of hand contours defects. Journal of Automation and Information Sciences, 48(4)(2016):90-98. doi:10.1615/JAutomatInfScien.v48.i4.80

[6] J. Smith, N. Ravi. The Architecture of Virtual Machines. Computer. IEEE Computer Society. 38 (5) (2005): 32–38.

[7] L. Prasuhn, Y. Oyamada, Y. Mochizuki, H. Ishikawa. A hog-based hand gesture recognition system on a mobile device. In 2014 IEEE International Conference on Image Processing (ICIP), pages 3973-3977. IEEE, 2014.

[8] N. H. Dardas, N. D. Georganas. Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. IEEE Transactions on Instrumentation and measurement, 60(11) (2011): 3592-3607.

[9] I.V. Krak, G.I. Kudin, A.I. Kulias. Multidimensional Scaling by Means of Pseudoinverse Operations, Cybernetics and Systems Analysis, 55(1) (2019):22-29. doi: 10.1007/s10559-019-00108-9

[10] O. Kopuklu , N. Kose, G. Rigoll. Motion fused frames: Data level fusion strategy for hand gesture recognition. arXiv preprint arXiv:1804.07187, 2018.

[11] P. Molchanov, S. Gupta, K. Kim, K. Pulli. Multi-sensor system for driver's hand-gesture recognition. In Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops, volume 1, pages 1-8. IEEE, 2015.

[12] P. Molchanov, S. Gupta, K. Kim, J. Kautz. Hand gesture recognition with 3d convolutional neural networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pages 1-7. June 2015.

[13] A. Krizhevsky, I. Sutskever, G.E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097-1105, 2012.

[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 1725-1732, 2014.

[15] J. Carreira, A. Zisserman. Quovadis, action recognition a new model and the kinetics dataset. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference, pages 4724-4733. IEEE, 2017.

[16] T. B. N. GmbH. The 20bn-jester dataset v1. https://20bn.com/datasets/jester, 2019.

[17] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size. arXiv preprint arXiv:1602.07360, 2016.

[18] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4510-4520. IEEE, 2018.

[20] X. Zhang, X. Zhou, M. Lin, J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6848–6856. IEEE, 2018.

[21] N. Ma, X. Zhang, H.-T. Zheng, J. Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. arXiv preprint arXiv:1807.11164, 5, 2018.

[22] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang. Searching for MobileNetV3. axXiv: 1905.02244, 5, 2019

[23] Eng-Jon Ong et al. Sign language recognition using sequential pattern trees. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE. 2012, pp. 2200-2207

[24] Tensorflow framework documentation [https://www.tensorflow.org/api/]

[25] K. Hara, H. Kataoka, Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pages 18-22, 2018.

[26] J. Hu, L. Shen, G. Sun. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132-7141, 2018.

[27] K. He, X. Zhang, S. Ren, J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770-778, 2016.