Student: Konstantin Kobranov

Matriculation number: 967086

Program: Data science and economics

Email: konstantin.kobranov@studenti.unimi.it

Investigation of the performance of convolutional neural networks in sound recognition

1. Introduction

An urban sound classification is a vital area in audio informatics. Thus, modern classification techniques are used in various fields such as smart cities: noise monitoring [1] and soundscape assessment [2]; robot navigation; acoustic monitoring of natural environment [3] etc.

In comparison, for instance, with speech recognition sound classification in an urban environment may be really challenging as require preprocessing information obtained from different sources with some noise level. As a result, the application of machine learning algorithms requires great precision and patient from the researcher.

One of the main issues concerning sound classification relates to the feature extraction step. The first approach is based on so-called Mel-Frequency Cepstral Coefficients (MFCC). MFCC represents time-series obtained by applying several transforms, including Fourier transform, to the original signal. The second one implies using the so-called spectrogram representation of the original signal which is the distribution of the spectrum of frequencies across time.

Various machine learning algorithms were used to solve the described task. Hence, in the work [4] the following techniques were used: k-NN, random forest, SVM and a baseline majority vote classifier using MFCC. For these purposes, the *UrbanSound8K* dataset was used. The experiments were run using 10-fold cross-validation, and the highest accuracy was obtained for SVM (approximately 0.7). In paper [5] the authors applied a deep convolutional neural network (CNN) on log-mel spectrograms with several data augmentation techniques on the same dataset. Using the same cross-validation approach as in the previous paper the highest achieved test accuracy is 0.79. The paper [6] illustrates the CNN model on the log-scaled mel-spectrograms. The author proposed a technique to extract features like 50% overlapping segments of the original spectrograms. The model was evaluated in a 10-fold for the *UrbanSound8K* dataset. Thus, the highest accuracy for the proposed model is approximately 0.74. The work [7] introduces a modified approach which implies using the combination of the features, namely MFCC, spectrograms and Cross Recurrence Plot (CRP). As a result, the models' input (for AlexNet and GoogleNet) is an RGB image of the size (256, 256, 3). The test accuracy using 10-fold-cross-validation for AlexNet and GoogleNet are 0.92 and 0.93 respectively.

The goal of the present research is to investigate the performance of convolutional neural networks in sound recognition.

The paper is organized as follows. Firstly, in the section "Methodology" the used models, dataset and feature extraction method are described. Secondly, in the section "Results and Discussions" the influence of the sampling rate on the model performance, the hyperparameter tuning for the chosen models and the comparison of the chosen models are shown. Finally, in the "Conclusions" the relevant conclusions on the work were made.

2. Methodology

(a) Models

In the present work, the following models were used: AlexNet [8] and CNN [5, 9] with a little modification, namely the input size was changed to (128, 128, 1).

The AlexNet model consists of eight layers, namely five convolutional layers and three fully connected layers. The first convolutional layer contains 96 kernels of size (11, 11) followed by a max-pooling layer of size (2, 2). The second convolutional layer contains 256 kernels of size (5, 5) followed by a max-pooling layer of size (2, 2). The third convolutional layer with 512 kernels of size (3, 3) has a zero-padding layer with padding (1,1) and a max-pooling layer of size (2, 2). The fourth and the fifth convolutional layers contain 1024 kernels of size (3, 3) with a zero-padding layer with padding (1, 1). The fifth layer is followed by the max pooling layer of size (2, 2). The sixth layer (the first fully connected layer) has 3072 units. The seventh layer has 4096 units. Additionally, batch normalization and dropout with a rate equal to 0.5 are used. A ReLU activation function was used.

The CNN model consists of three convolutional layers and two fully connected layers. The first layer is convolutional and contains 24 kernels of size (5, 5). The next two convolutional layers contain 48 kernels of size (5, 5). The first two layers have a max-pooling layer of size ((4, 2), (4, 2)). The fourth layer which is the first fully connected layer has 64 units. All fully connected layers use dropout with a rate equal to 0.5. A ReLU activation function was used.

In both models, an Adam optimizer was used. Additionally, Stochastic Gradient Descent (SGD) was used for AlexNet with parameters taken from the work [7].

(b) Experiment setup

In the present work, the *UrbanSound8K* dataset [10] was used. The dataset consists of 8732 audio signals. The duration of signals varies but do not exceed 4 seconds on average. The dataset contains information about 10 urban sound classes: "air conditioner", "car horn", "children playing", "dog bark", "drilling", "engine idling", "gunshot", "jackhammer", "siren" and "street music". Moreover, "UrbanSound8k.csv" containing meta-data is presented. Finally, sound files were divided into 10 predefined folds.

In the present work magnitude mel-spectrograms was used as features since they guarantee the highest performance in comparison with log-scaled mel-spectrogram [9]. The spectrograms were extracted from the raw data using *librosa*[1] library ver. 0.8.1. The number of mel bands was set equal to 128. Due to the variation in the length of the sample, all spectrograms were cast to a length equal to 128. All samples were loaded in a mono regime. As a result, the shape of the input data is (128, 128, 1). Different sampling rates, namely 44100 Hz, 22050 Hz, 11025 Hz and 5512.5 Hz was used and their influence on the model performance was studied. All input data was standardized [9]:

$$\mu = \frac{1}{T}\frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N} x_t^{(n)} \tag{1}$$

$$\sigma = \sqrt{\frac{1}{T}\frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N}(x_t^{(n)} - \mu)^2} \tag{2}$$

$$X_{norm} = \frac{X-\mu}{\sigma} \tag{3}$$

[1] librosa.org

Where X is the input sequence; $x_t^{(n)}$ is the value of the *n*-th feature at time *t*; *N* is a number of features; and *T* is a number of time steps.

The training of the models was performed on the folds: 1, 2, 3, 4, 6 and testing on the folds: 5, 7, 8, 9, 10. Thus, the performance of the models was evaluated as the average of the accuracies obtained on the test folds.

The *Google Colab* environment[2] with *Keras API*[3] ver. 2.6.0 were used.

The code for the present research can be found here[4].

3. Results and discussion

3.1. Influence of the sampling rate

In this section, the influence of the sampling rate on the model performance was studied. The following sampling rates were observed: 44100 Hz, 22050 Hz, 11025 Hz and 5512.5 Hz. The CNN model described above was used to conduct the experiment. The number of epochs and the batch size is equal to 64. The Adam optimizer with a learning rate equal to 0.001 was used.

Fig. 3 illustrates the dependence of the value of a loss on the last epoch on sampling rate.

Fig. 4 illustrates the dependence of accuracy (orange curve) and loss (blue curve) on the number of epochs on the validation set for sampling rate equals 44100 Hz (on the left) and sampling rate equals 11025 Hz (on the right).

It can be seen that the dependence of the loss on the validation sample has non-linear behavior. An increase in sampling rate leads to an increase in validation loss (Fig. 3). Indeed, at a high sampling rate (44100 Hz) slight overfitting can be observed i.e., the value of the validation loss starts increasing after approximately 25 epochs (Fig. 1 the blue curve on the left) whereas at a low sampling rate (11025 Hz) the value of the validation loss decreases to a value approximately equal to 0.79 and continues to oscillate around it (Fig. 2 the blue curve on the right).

From the results obtained, the lowest value of the validation loss is reached when the sampling rate is equal to 11025 Hz. As a result, the value of the sampling rate equals 11025 Hz will be used in further experiments.

[2] colab.research.google.com

[3] keras.io

[4] github.com/kkonstantin182/urban-sound-classification-project

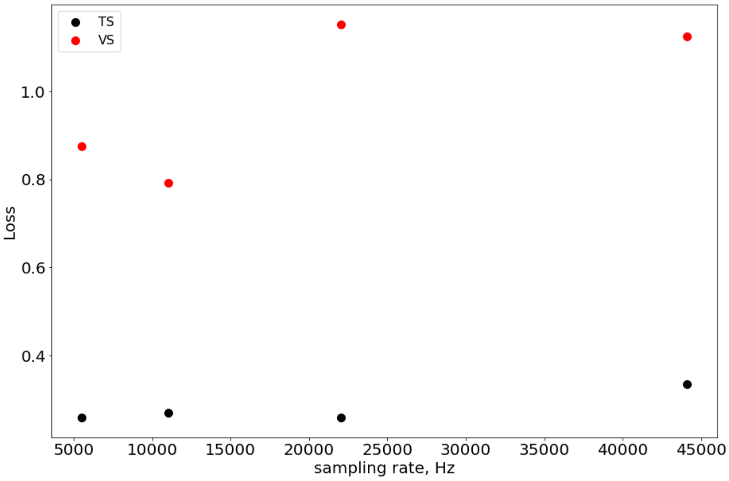


Fig. 3 Dependence of the value of a loss on the last epoch on sampling rate.

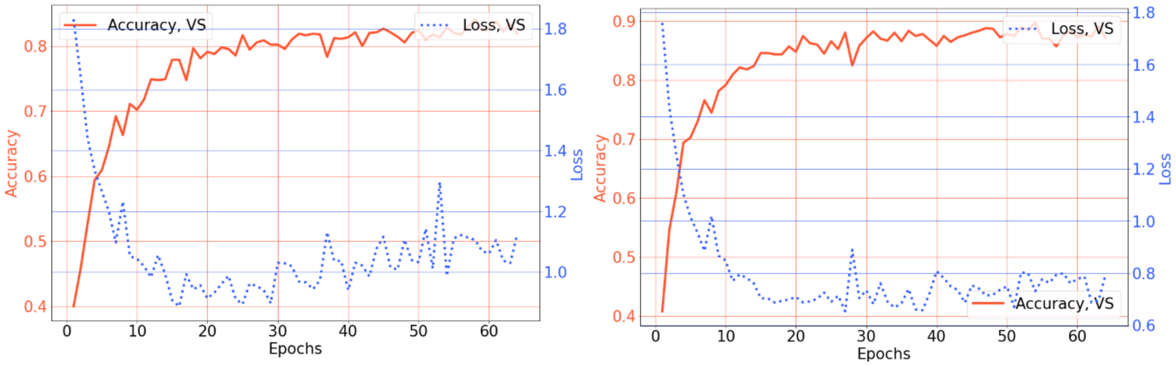


Fig. 4 Dependence of accuracy (orange curve) and loss (blue curve) on the number of epochs on the validation set for sampling rate equals 44100 Hz (on the left) and sampling rate equals 11025 Hz (on the right).

3.2. CNN

In this section performance of the described above CNN model will be evaluated and the following hyperparameters: the learning rate and the batch size will be tuned.

(a) Influence of the learning rate

The following values of the learning rate will be observed: 0.0001, 0.001, 0.01. The number of epochs and the batch size is equal to 64. The Adam optimizer was used.

Fig. 5 shows the dependence of the value of the loss on the validation set on the value of the logarithm of the learning rate.

Fig. 6 demonstrates the dependence of accuracy (orange curve) and loss (blue curve) on the number of epochs on the validation set for the learning rate equals 0.00001 (A), 0.0001 (B), 0.001 (C), 0.01 (D).

It can be seen that in the range from 0.00001 to 0.01 the dependence of validation loss on the learning rate has quasilinear behavior. Firstly, as the learning rate increase, the validation loss decreases. Then, the saturation area from the value of the logarithm of the learning rate approximately equals 10^{-4} to approximately 10^{-3} is observed. Finally, the validation loss starts increasing as the learning rate increases (Fig. 5, red curve). Indeed, when the value of the learning rate is low (0.00001) the validation loss does not reach the saturation area (Fig. 6 A), whereas when the value of the learning rate is high (0.01) the validation loss starts increasing after approximately 60 epochs (Fig. 6 D).

From the results obtained, the optimal value of the learning rate is 0.0001 as it guarantees the lowest value of the validation loss on the given interval of the learning rate. Thus, the value of the learning rate equal to 0.0001 will be used in further experiments for the given model.

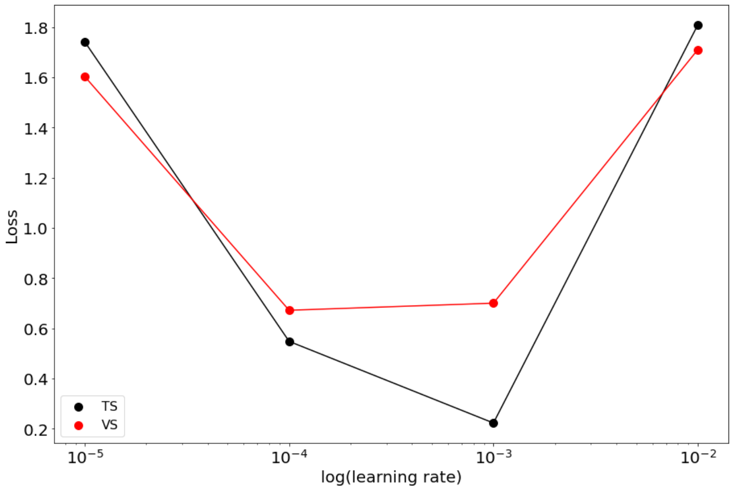
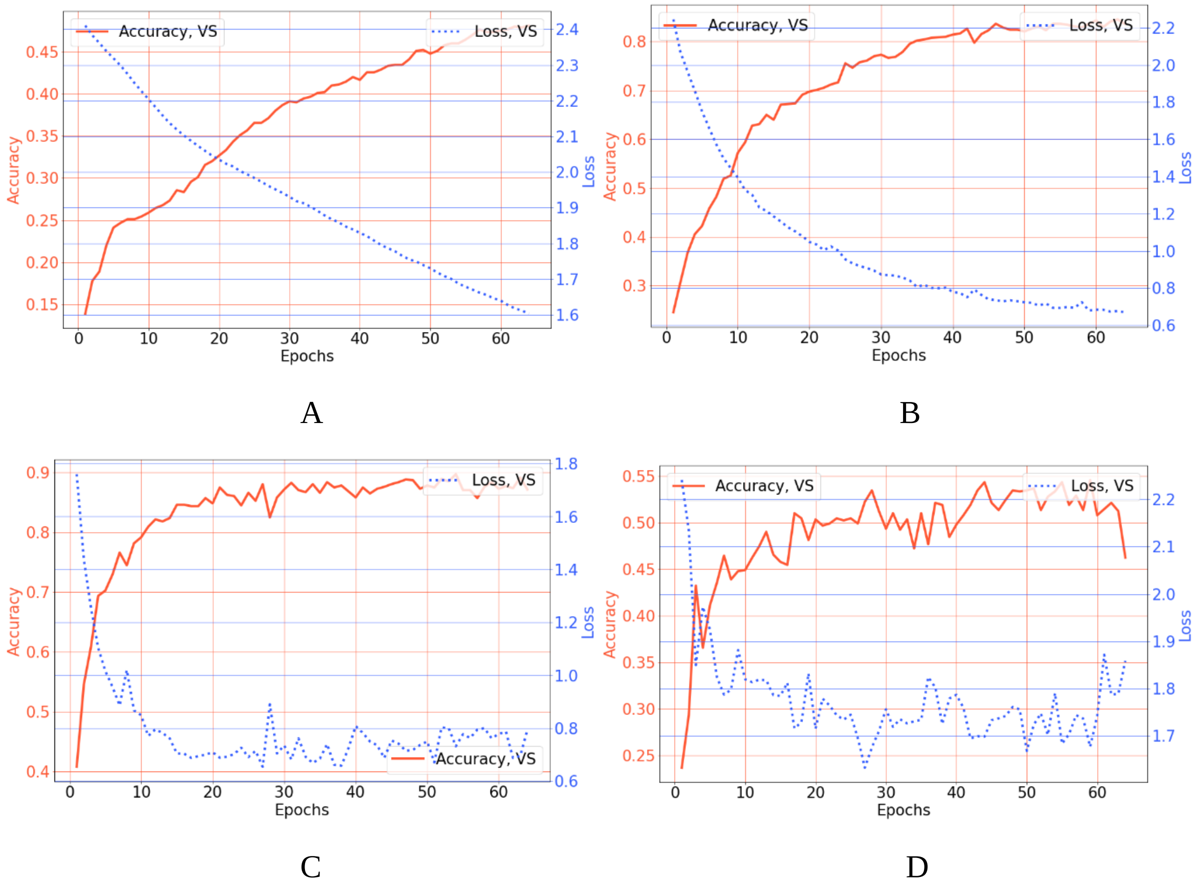


Fig. 5 Dependence of the value of a loss on the validation set on the value of the logarithm of the learning rate.



A

B

C

D

Fig. 6 Dependence of accuracy (orange curve) and loss (blue curve) on the number of epochs on the validation set for the learning rate equal to 0.00001 (A), 0.0001 (B), 0.001 (C) and 0.01 (D).

(b) Influence of the batch size

The following values of the batch size will be observed: 8, 16, 32, 64 and 128. The number of epochs is equal to 64. The Adam optimizer was used with a learning rate equal to 0.0001 was used.

Fig. 7 depicts the dependence of the value of a loss on the validation set on the value of the batch size.

Fig. 8 portrays the dependence of accuracy (orange curve) and loss (blue curve) on the number of epochs on the validation set for the batch size equals 8 (A), 16 (B) and 128(C).

It can be observed that the dependence of the value of a loss on the validation set on the value of the batch size has non-linear behavior. The validation loss decreases on the interval of batch size from 8 to 16. Then it grows almost linearly from 16 to 128.

As a consequence, the value of the batch size equal to 16 should be used in further research since it is the point where validation loss reaches its lowest value.

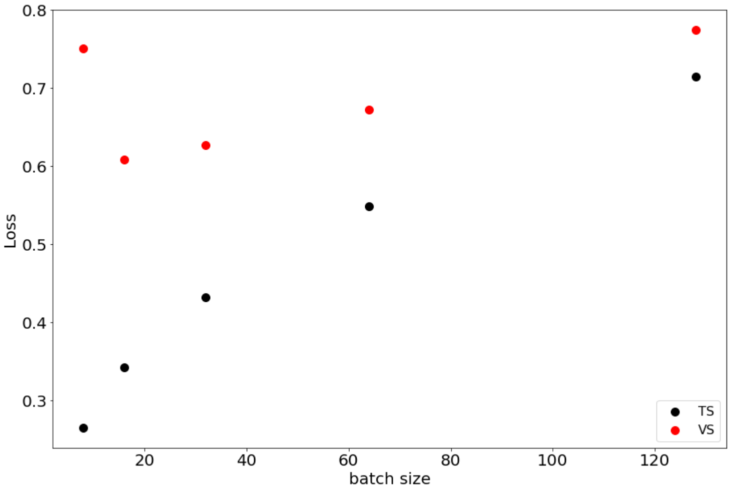
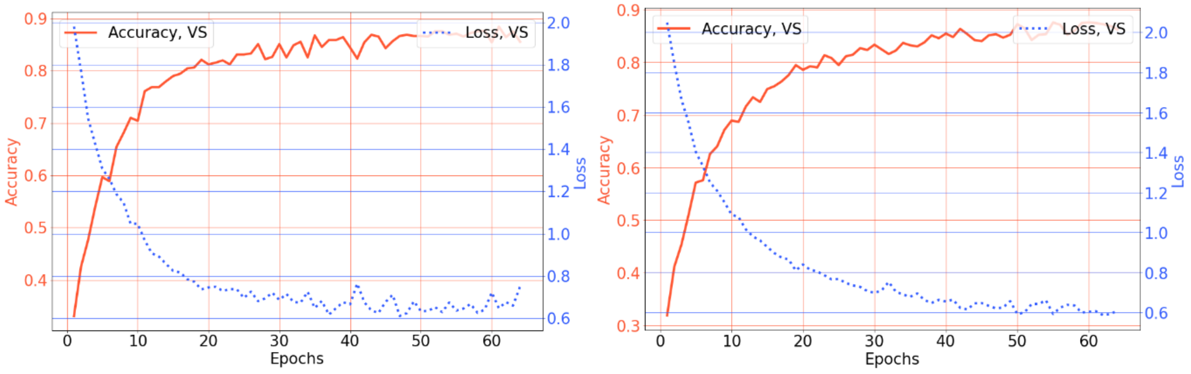
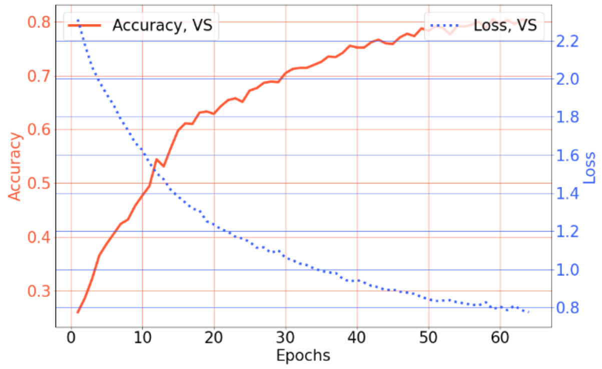


Fig. 7 Dependence of the value of a loss on the validation set on the value of the batch size.



A

B



C

Fig. 8 Dependence of accuracy (orange curve) and loss (blue curve) on the number of epochs on the validation set for the batch size equal to 8 (A), 16 (B) and 128(C).

3.3.AlexNet

In this section performance of the described above AlexNet model will be evaluated and the following hyperparameters: the learning rate and the batch size will be tuned.

(a) Influence of the learning rate

The following values of the learning rate will be observed: 0.00001, 0.0001, 0.001, 0.01. The number of epochs and the batch size is equal to 64. The Adam optimizer was used.

Fig. 9 shows the dependence of the value of a loss on the validation set on the value of the learning rate.

Fig. 10 illustrates the dependence of accuracy (orange curve) and loss (blue curve) on the number of epochs on the validation set for the learning rate equals 0.00001 (A), 0.0001 (B), 0.001 (C) and 0.01 (D).

It can be seen that the dependence of the validation loss on the learning rate has non-linear behavior. Firstly, the validation loss decreases as the learning rate increases up to approximately 0.4 which corresponds to the value of the learning rate equal to 0.001 (Fig. 10 C). Then, the validation loss increases with increasing the learning rate up to approximately 0.7 (Fig. 10 D). Additionally, from Fig. 10 we see that in comparison with the CNN model, the AlexNet model seems not to be suffering from overfitting.

From the results obtained, it can be concluded that the optimal value of the learning rate is equal to 0.001 as at this value the smallest validation loss value is reached. As a result, the value of the learning rate equals 0.001 will be used in further experiments.

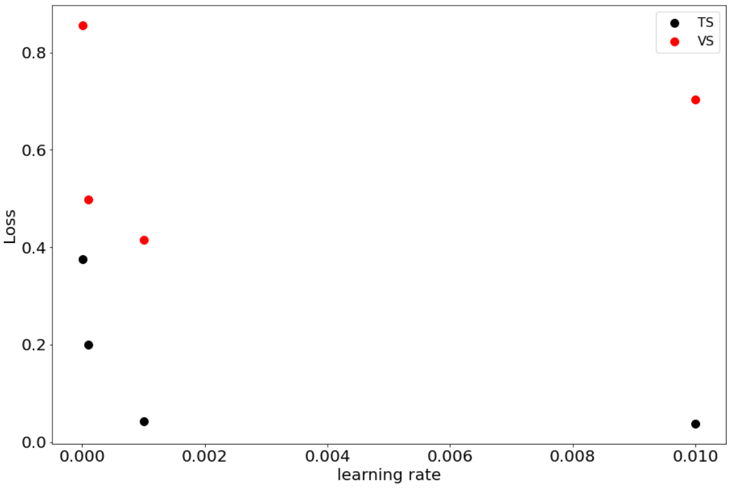


Fig. 9 Dependence of the value of a loss on the validation set on the value of the learning rate.

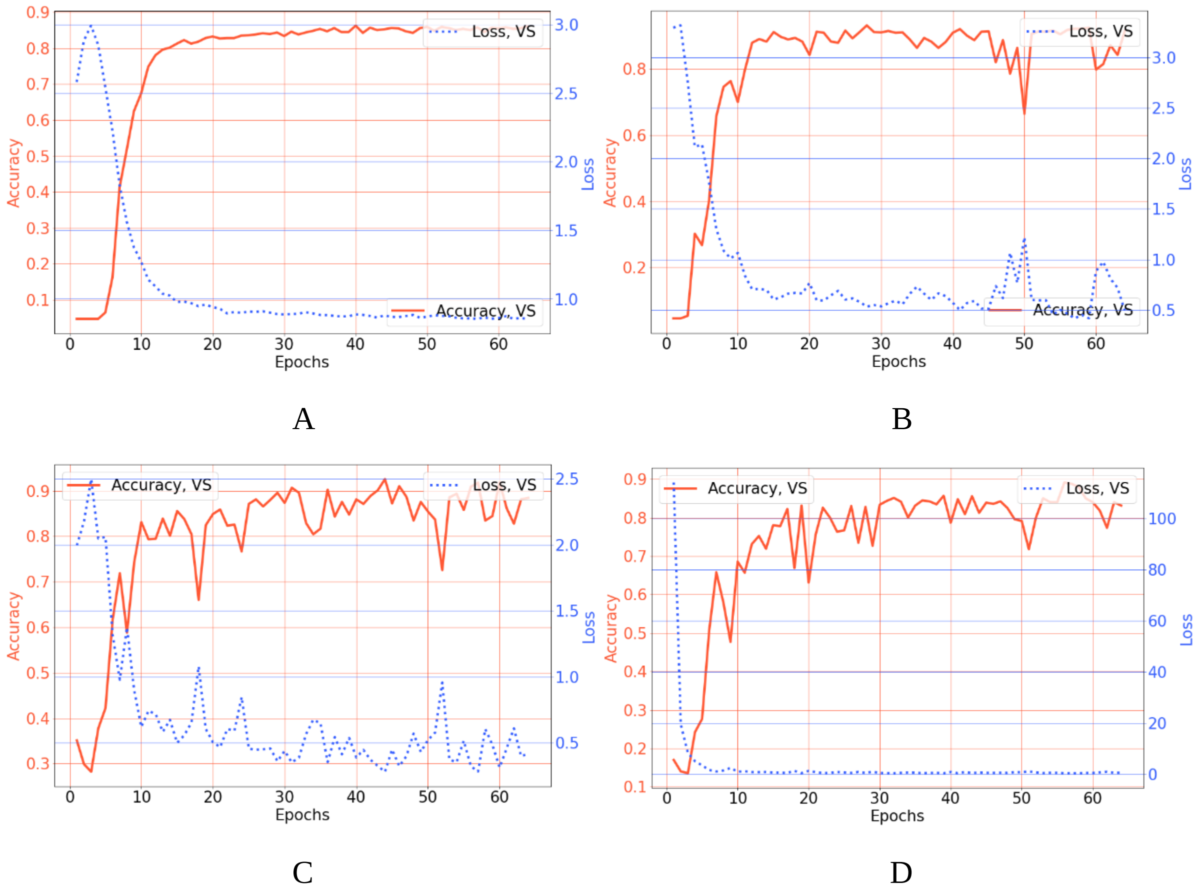


Fig. 10 Dependence of accuracy (orange curve) and loss (blue curve) on the number of epochs on the validation set for the learning rate equals 0.00001 (A), 0.0001 (B), 0.001 (C) and 0.01 (D).

(b) Influence of the batch size

The following values of the batch size will be observed: 16, 32, 64 and 128. The number of epochs equals 64. The Adam optimizer was used with a learning rate equal to 0.001 was used.

Fig. 11 illustrates the dependence of the value of a loss on the validation set on the value of the batch size

Fig. 12 portrays the dependence of accuracy (orange curve) and loss (blue curve) on the number of epochs on the validation set for the batch size equals 8 (A), 16 (B), 32 (C) and 128 (D).

It can be observed that the dependence of the value of a loss on the validation set on the value of the batch size has non-linear behavior. Thus, the value of the loss decreases as the batch size increases up to approximately 0.35 which corresponds to the value of the batch size equals 32. Then, the validation loss starts increasing with the batch size increases and reaches the value of approximately equals 0.8 which corresponds to the value of the batch size equals 128.

From the results obtained, the value of the batch size equals 32 should be used as it guarantees the lowest validation loss.

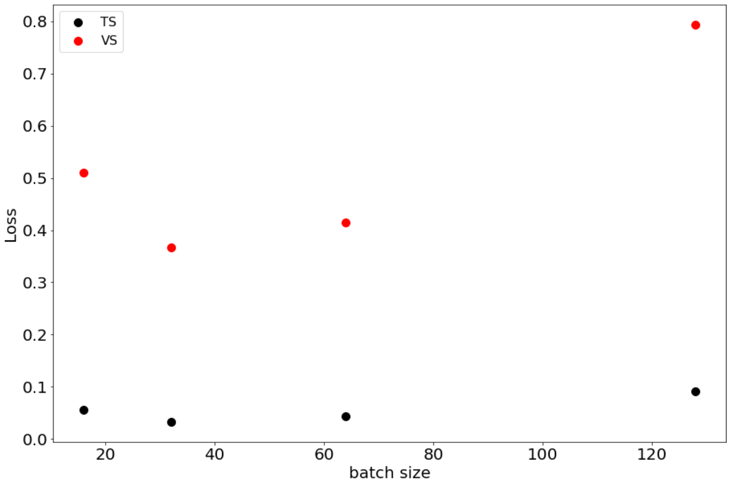
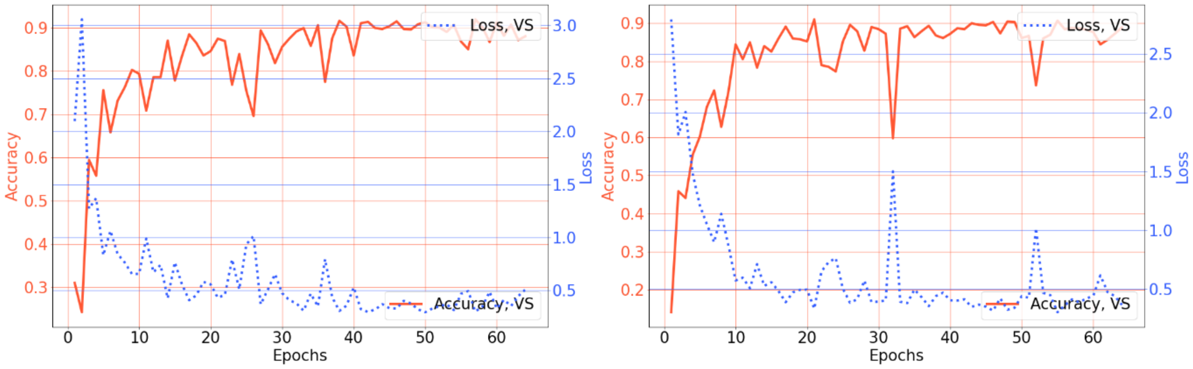
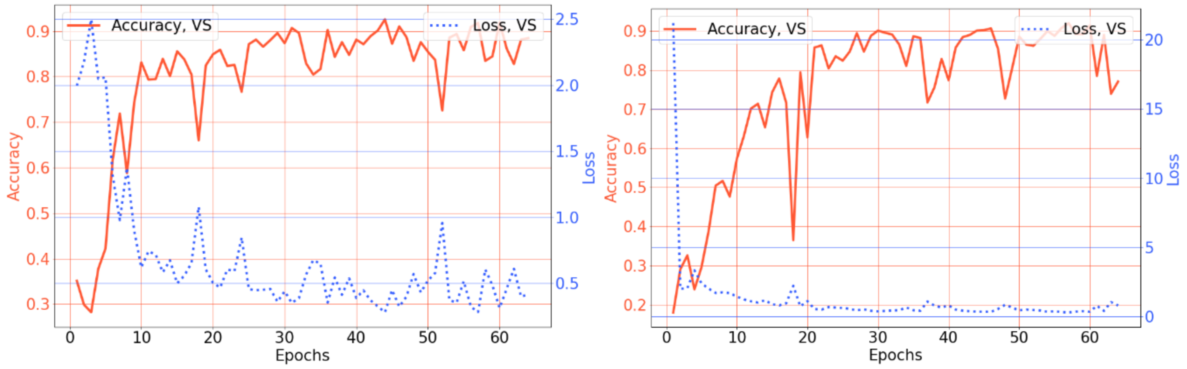


Fig. 11 Dependence of the value of a loss on the validation set on the value of the batch size.



A

B



C D

Fig. 12 Dependence of accuracy (orange curve) and loss (blue curve) on the number of epochs on the validation set for the batch size equal to 8 (A), 16 (B), 32 (C) and 128 (D).

3.4.Model comparison

The CNN and the AlexNet models were compared in terms of the average test accuracy with the optimal values of hyperparameters. Thus, the CNN model was tested with the value of learning rate equals 0.0001 and the values of the batch size equal 16. The AlexNet model was tested with the value of learning rate equals 0.001 and the values of the batch size equal to 32. Additionally, the AlexNet model (denoted as "AlexNet*" in Table 1) with the value of the hyperparameters taken from the article [7] was included in the analysis (an optimizer: Stochastic Gradient Descent, learning rate change policy: exponential decay with the base learning rate equals 0.01 and the decay rate equals 0.95).

Table 1 demonstrates test accuracy obtained on the folds: 5, 7, 8, 9 and 10, as well as, average accuracy for each model.

From the results obtained, it can be concluded that the highest test accuracy is achieved using the CNN model. The CNN model outperforms the AlexNet model with hyperparameters taken from the work [7] by 4%. The AlexNet model with the value of hyperparameters optimized during present research shows the lowest result i.e., 0.61±0.02 in comparison with 0.63±0.03.

Table 1

Model\Acc.	Fold 5	Fold 7	Fold 8	Fold 9	Fold 10	Avg. acc.
CNN	0.6731	0.6480	0.6489	0.6446	0.6619	0.66±0.01
AlexNet	0.6239	0.6193	0.6129	0.5797	0.6308	0.61±0.02
AlexNet*	0.6763	0.6229	0.6290	0.5760	0.6392	0.63±0.03

4. Conclusions

As a result of the research done, the CNN model proposed by authors of the work [5] and the AlexNet model proposed by the authors of the work [8] were studied. The models used in this work have been optimized in terms of the hyperparameters used. The following conclusions have been drawn:

- For the given dataset the sampling rate equals 11025 Hz should be used as the lowest value of the validation loss is reached using this value of the sampling rate.
- For the given dataset the CNN model outperforms the AlexNet model by 4% in terms of the average accuracy. Additionally, using the CNN model is highly recommended as it has a smaller number of parameters and, as a result, require less training time.

5. Declaration

I/We declare that this material, which I/We now submit for assessment, is entirely my/our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my/our work. I/We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me/us or any other person for assessment on this or any other course of study.

6. References

[1] Huang, Z., Liu, C., Fei, H., Li, W., Yu, J., & Cao, Y. (2020). Urban sound classification based on 2-order dense convolutional network using dual features. *Applied Acoustics*, *164*, 107243. https://doi.org/10.1016/j.apacoust.2020.107243

[2] Mesaros, A., Heittola, T., & Virtanen, T. (2016). TUT database for acoustic scene classification and sound event detection. *European Signal Processing Conference*, *2016-November*, 1128–1132. https://doi.org/10.1109/EUSIPCO.2016.7760424

[3] Xie, J., Colonna, J. G., & Zhang, J. (2021). Bioacoustic signal denoising: a review. *Artificial Intelligence Review*, *54*(5), 3575–3597. https://doi.org/10.1007/s10462-020-09932-4

[4] Justin Salamon; Christopher Jacoby; Juan Pablo Bello. (2014). Urban Sound Datasets. *MM '14 Proceedings of the 22nd ACM International Conference on Multimedia*, (3), 1041–1044. Retrieved from http://serv.cusp.nyu.edu/projects/urbansounddataset/

[5] Salamon, J., & Bello, J. P. (2017). Deep Convolutional Neural Netwsorks and Data Augmentation for Environmental Sound Classification. *IEEE Signal Processing Letters*, *24*(3), 279–283. https://doi.org/10.1109/LSP.2017.2657381

[6] Piczak, K. J. (2015). Environmental sound classification with convolutional neural networks. *IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, *2015-Novem*. https://doi.org/10.1109/MLSP.2015.7324337

[7] Boddapati, V., Petef, A., Rasmusson, J., & Lundberg, L. (2017). Classifying environmental sounds using image recognition networks. *Procedia Computer Science*, *112*, 2048–2056. https://doi.org/10.1016/j.procs.2017.08.250

[8] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*, 84–90.

[9] Lezhenin, I., Bogach, N., & Pyshkin, E. (2019). Urban sound classification using long short-term memory neural network. *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019*, *18*, 57–60. https://doi.org/10.15439/2019F185

[10] Justin Salamon; Christopher Jacoby; Juan Pablo Bello. (2014). Urban Sound Datasets. *MM '14 Proceedings of the 22nd ACM International Conference on Multimedia*, (3), 1041–1044. Retrieved from http://serv.cusp.nyu.edu/projects/urbansounddataset/