

Transfer Learning Tuning Utilizing Grey Wolf Optimizer for Identification of Brain Hemorrhage from Head CT Images

Grega Vrbančič
University of Maribor, Faculty
of Electrical Engineering and
Computer Science
Koroška cesta 46
SI-2000 Maribor, Slovenia
grega.vrbancic@um.si

Milan Zorman
University of Maribor, Faculty
of Electrical Engineering and
Computer Science
Koroška cesta 46
SI-2000 Maribor, Slovenia
milan.zorman@um.si

Vili Podgorelec
University of Maribor, Faculty
of Electrical Engineering and
Computer Science
Koroška cesta 46
SI-2000 Maribor, Slovenia
vili.podgorelec@um.si

ABSTRACT

Most commonly, diagnosing the brain hemorrhage - a condition caused by a brain artery busting and causing bleeding is done by medical experts identifying such pathologies from the computer tomography (CT) images. With great advancements in the domain of deep learning, utilizing deep convolutional neural networks (CNN) for such tasks has already proven to achieve encouraging results. One of the major problems of using such an approach is the need for big labeled datasets to train such deep architectures. One of the efficient techniques for training CNNs with smaller datasets is transfer learning. For the efficient use of transfer learning, many parameters are needed to be set, which are having a great impact on the classification performance of the CNN. Most of those parameters are commonly set based on our previous experience or by trial and error. The proposed method addresses the problem of tuning the transfer learning technique utilizing the nature-inspired, population-based metaheuristic Grey Wolf Optimizer (GWO). The proposed method was tested on a small head CT medical imaging dataset. The results obtained from the conducted experiments show that the proposed method outperforms the conventional approach of parameter settings for transfer learning.

Keywords

Convolutional Neural Network, Transfer Learning, Optimization, Biomedical images, Classification

1. INTRODUCTION

Most commonly used medical imaging technique to assess the severity of brain hemorrhage, also termed as a cerebral hemorrhage, intracranial hemorrhage or intracerebral hemorrhage is the computer tomography or shortly CT. As reported in [24], each year intracerebral hemorrhage (ICH) affects 2.5 per 10,000 people worldwide and is associated with high mortality that only 38% of ICH patients could survive over one year. Besides, more than 80% of people are suffering due to being born with a weak spot in their major brain arteries. However, the early diagnosis of the condition and receiving immediate and relevant treatment can be a lifesaver for the affected patient. Traditionally, the tools helping in diagnosing such conditions are CT images obtained from the CT scan, which are then examined by the expert such as an experienced doctor, who has the ability to identify important symptoms of the disease from the image

by a naked eye [3].

With the expansion of deep learning field and with the great achievements of deep convolutional neural networks (CNN) for the image and video recognition tasks [26, 27] are such approaches and methodologies also being used for addressing various medical areas such as medical image analysis [1] and classification [31, 12], biomedical signal segmentation [23] and detection of various human organ activities [30].

In recent studies [12, 4, 11], the authors have already addressed the problem of identifying various kinds of brain hemorrhages utilizing different kinds of more or less complex deep CNNs. However, the problem with the training of such deep CNN architectures remains the same. In order to achieve acceptable performance, the training of such networks requires a lot of resources in terms of time and processing power. Additionally, a big dataset of images, hand-labeled by experts is also required. Given the fact that such high-quality big datasets of biomedical images are hard to obtain, researchers are trying various approaches and techniques to overcome this problem. One of the most popular techniques for training deep CNNs on small datasets is transfer learning, which has already proven to achieve great results [4, 14]. But the transfer learning techniques also comes with the downsides. Most commonly, the biggest problems when utilizing the transfer learning approaches are finding out which and how many layers to fine-tune and how to set the training parameters for the fine-tuning of the CNN in order to obtain the acceptable outcome.

Based on the encouraging results of transfer learning technique being used to train CNNs for the task of classification of biomedical images and our previous experience on optimizing various training parameters [32], we set our goal to develop a method for an automatic optimization of transfer learning utilizing nature-inspired population-based Grey Wolf Optimizer (GWO) algorithm named GWOTLT.

The rest of the paper is organized as follows. In Section 2, we briefly describe methods which were used. In Section 3, we present the proposed GWOTLT method, while in Section 4 we describe the experimental setup of conducted experiments, the results of which are presented in Section 5. Conclusions and final remarks are gathered in Section 6.

2. METHODS

In this section, the methods utilized in our proposed GWOTLT method are briefly presented.

2.1 Convolutional Neural Network

In the 1980s, the CNNs were first presented in Fukushima's paper [10]. The author proposed a deep learning approach for visual recognition, called neocognitron, which was based on the hierarchical layers trained with the utilization of the stochastic gradient descent algorithm. The major breakthrough with CNNs occurred in 1998 with the LeCun's LeNet5 [17] proposed architecture which is considered to be one of the key factors that started the enormous expansion of the deep learning field.

Initially, the deep CNNs were defined as 2-dimensional constrained neural networks with alternating convolutional and subsampling or pooling layers which are fully connected at the end, combining three architectural ideas [17]:

- local receptive fields,
- shared weights, and
- spatial and temporal subsampling.

Most commonly the convolutional layer is composed of several so-called feature maps. Those feature maps are calculated with different weight vectors, which enable us to extract multiple features from each location. The results of the convolutional calculation are obtained from a convolutional operation performed between feature maps of the previous layer and convolution kernel of the current layer in addition to the activation function. A subsampling layer or pooling layer reduces the dimension of feature maps, while preserving the important extracted features, usually performing local averaging and subsampling. The fact, that extracted features' real locations are not important as long as their approximate positions relative to others remain the same, is making subsampling possible [17].

Although the researchers have through the years developed various complex CNN architectures which proven to be highly successful in the large-scale image and video recognition such as Krizhevsky's AlexNet [15], Szegedy's GoogleNet [27] and Simonyan's VGG16 [26], the challenges regarding image and video recognition still exist. Such major challenges are primarily the need for large datasets in order to train the CNNs and the time complexity of the training process.

2.2 Transfer Learning

One of the most popular approaches to address the time complexity of deep CNN training process as well as the problem of not having large dataset is known as a transfer learning. Transfer learning can be defined as the improvement of learning a new task through the transfer of knowledge from a related task that has already been learned. In machine learning terms, the transfer learning roughly translates to transferring the weights of already trained deep neural network model for one task, to the model tackling second related task [13]. Based on previous work [16, 2, 25], such

approaches work especially well if we have a small, insufficient dataset.

Transfer learning is most commonly used in two ways [2, 21]:

- Fine-tuning in which the weights of the pre-trained CNN base model are preserved (frozen) on some of the layers and fine-tuned (trained) in remaining layers of CNN.
- CNN as a feature extractor, where the general idea is to access features of any layers and using those encoded features to train a classifier of your choice.

Generally, the first (top) layers of the CNN preserve more abstract, generic features applicable to other tasks, while the layers closer to the bottom provide more specific features that can benefit from fine-tuning as they will be adjusted specifically for the targeted task. For the fine-tuning approach to transfer learning, there is no general recipe or rule to follow on selecting which layers to tune and which ones to preserve as they are. Also, another challenge utilizing the fine-tuning approach is deciding how many layers to add to the bottom of the pre-trained convolutional base, and which optimizer and learning rate to use in the process of fine-tuning.

2.3 Grey Wolf Optimizer

In recent years, swarm intelligence and bio-inspired algorithms for solving the optimization problems are quite popular and proven to be very efficient in solving real-world problems [9].

One of the most popular representatives of such optimization algorithms is a Grey Wolf Optimizer or simply GWO [19]. The inspiration of GWO is adapted from a strict leadership hierarchy and hunting mechanisms of grey wolves (*Canis lupus*). The grey wolf leadership hierarchy is divided into four dominance groups, i.e. alpha, beta, delta and, omega. Besides the leadership hierarchy, group hunting is also an interesting social behavior of grey wolves. As defined by authors in [20] main phases of grey wolf hunting are as follows [19]:

- Tracking, chasing and approaching the prey.
- Pursuing, encircling, and harassing the prey until it stops moving.
- Attack towards the prey.

The GWO algorithm implementation is mathematically modeling the mentioned hunting technique and the social hierarchy in order to perform optimization. The basic pseudo-code of GWO algorithm is presented in Algorithm 1.

3. PROPOSED METHOD

The basic concept of our proposed method for tuning of transfer learning approach based on the GWO algorithm, named as GWOTLT is presented in Figure 1. The GWO algorithm is used to find the optimal parameters for the

Algorithm 1 Pseudo code of the GWO algorithm.

```
1: Initialize the grey wolf population  $X_i$  ( $i = 1, 2, \dots, n$ )
2: Initialize  $a$ ,  $A$ , and  $C$ 
3: Calculate the fitness of each search agent
4:  $X_\alpha$  = the best search agent
5:  $X_\beta$  = the second best search agent
6:  $X_\delta$  = the third best search agent
7: while  $i < \text{Maximum iterations}$  do
8:   for each search agent do
9:     Update the position of the current search agent
10:  end for
11:  Update  $a$ ,  $A$ , and  $C$ 
12:  Calculate the fitness of all search agents
13:  Update  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$ 
14:   $i = i + 1$ 
15: end while
16: return  $X_\alpha$ 
```

fine-tuning transfer learning process. In our case, the goal is to find a number of neurons in the last fully connected layer, dropout probability of dropout layer and the most suitable optimizer and learning rate value.

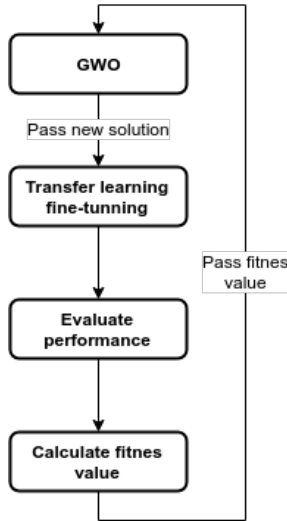


Figure 1: The conceptual diagram of the proposed GWOTLT method.

Given the number of optimized parameters for fine-tuning of the transfer learning process, the GWOTLT is producing the solution with the dimension of 4. The individuals of GWOTLT produced solutions are presented as real-valued vectors:

$$\mathbf{x}_i^{(t)} = (x_{i,0}^{(t)}, \dots, x_{i,n}^{(t)}), \quad \text{for } i = 0, \dots, Np - 1, \quad (1)$$

where each element of the solution is in the interval $x_{i,1}^{(t)} \in [0, 1]$.

In next step, the real-valued vectors (solutions) are mapped as defined in equations 2, 3, 4 and 5, where y_1 presents the number of neurons in last fully connected layer, y_2 dropout

probability, y_3 optimization function and y_4 learning rate. Each y_1 value is mapped to the particular member of the population $N = \{64, 128, 256, 512, 1024\}$ according to the members position in the population, which represents a group of available numbers of neurons in last fully connected layer. All of the y_3 values are mapped to the specific member of population $O = \{adam, rmsprop, sgd\}$, which represents a group of available optimizer functions, while each y_4 values are mapped to the member of population $L = \{0.001, 0.0005, 0.0001, 0.00005, 0.00001\}$, which represents a group of learning rate choices.

$$y_1 = \begin{cases} \lfloor x[i] * 5 + 1 \rfloor; y_1 \in [1, 5] & x[i] < 1 \\ 5 & \text{otherwise,} \end{cases} \quad (2)$$

$$y_2 = x[i] * (0.9 - 0.5) + 0.5; y_2 \in [0.5, 0.9] \quad (3)$$

$$y_3 = \begin{cases} \lfloor x[i] * 3 + 1 \rfloor; y_3 \in [1, 3] & x[i] < 1 \\ 3 & \text{otherwise,} \end{cases} \quad (4)$$

$$y_4 = \begin{cases} \lfloor x[i] * 5 + 1 \rfloor; y_4 \in [1, 5] & x[i] < 1 \\ 5 & \text{otherwise,} \end{cases} \quad (5)$$

To evaluate each solution produced by GWOTLT the fitness function was defined as follows:

$$f(x) = 1 - AUC(x) \quad (6)$$

where $f(x)$ is the fitness value for solution x and the $AUC(x)$ is an area under the ROC curve calculated on test split of the search dataset sub-sample.

4. EXPERIMENT SETUP

To evaluate the performance of our proposed method, we conducted two experiments. The experimental settings, dataset, evaluation methods and metrics used are in-depth presented in the following subsections.

The proposed method was implemented in Python programming language with the following external libraries: Numpy [28], Pandas [18], scikit-learn [22], NiaPy [29], Keras [5] and Tensorflow [7].

All of the conducted experiments were performed using the Intel Core i7-6700K quad-core CPU running at 4 GHz, 64GB of RAM, and three Nvidia GeForce Titan X Pascal GPUs each with dedicated 12GB of GDDR5 memory, running the Linux Mint 19 operating system.

4.1 Dataset

Given the task - identification of brain hemorrhage from CT images, we used a publicly available dataset of manually

collected head CT scan images called Head CT - hemorrhage [8]. The dataset contains in total of 200 images of various sizes. Of those 200 images, half of them are images of normal head CT slides without any brain pathologies, and the other half are the images containing some kind of brain hemorrhage. Also, each image is collected from a different person.

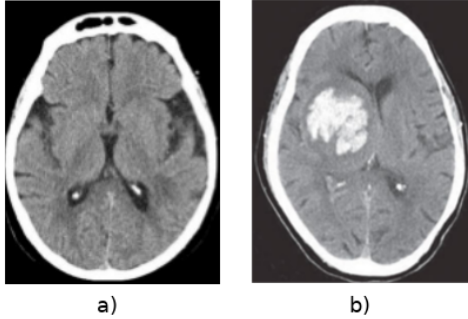


Figure 2: Example images of head CT scans, where a) represents normal head CT scan image. while b) represents the head CT scan image with brain hemorrhage present.

4.2 Grey Wolf Optimizer settings

To initialize the GWO algorithm, tackling the problem of finding the best suitable set of parameters to achieve the best performance of transfer learning fine-tuning, the GWO parameter settings presented in Table 1 were used.

Parameter	Value
Dimension of the problem	4
Population size	10
Number of function evaluations	50
Lower bound	0.0
Upper bound	1.0

Table 1: The initial GWO parameter settings.

4.3 Baseline Convolutional Neural Network

For the convolutional base of our proposed method, we utilized the VGG16 [26] CNN architecture presented in Figure 3, pre-trained on the imagenet [6] dataset. As we can observe from the figure, the VGG16 CNN is comprised of 5 convolutional blocks, which together form a convolutional base. At the bottom of the convolutional base a flatten layer, two fully connected layers and one fully-connected layer with softmax activation function forming a classifier layer are chained. By default, VGG16 CNN on the input receives an image of size 224 x 224 pixels and at the bottom classifies fed images into 1000 classes, while each of the convolutional layers of VGG architecture utilizes the ReLU activation function.

Performing the transfer learning based on the VGG16 CNN convolutional base, we have persisted the top four convolutional blocks and enabled for fine-tuning only last convolutional block. At the bottom of this convolutional base, we have then chained a flatten layer, a dropout layer, fully connected layer and classifier with softmax activation function,

classifying images into two target classes - images with and images without brain hemorrhage present.

For the baseline experiments, we have set the parameters which are we optimizing to the values presented in Table 2.

Parameter	Value
Number of neurons on the last fully connected layer	256
Dropout probability	0.5
Optimizer function	RMSprop
Learning rate	10^{-5}

Table 2: Baseline experiment parameter settings for transfer learning fine tuning.

With the presented parameter settings, we trained the CNN for 50 epochs utilizing an efficient mini-batch training, with batch size set to 32. As presented in the dataset section, the collected image sizes vary from 100 x 100 px to 300 x 300 px, thus we have decided to resize all images to the VGG16 default input size of 224 x 224 px.

4.4 GWOTLT settings

As presented in the previous section, the GWOTLT fine-tuning transfer learning parameters are set based on the produced GWO solution. The overall architecture of the convolutional base and the appended classification layers at the bottom are the same as in the baseline experiment. Due to the iterative nature of our proposed method, we had to split the given train set in ratio 80:20, where we used the larger subset for training different GWOTLT produced solutions and evaluating them - calculating the AUC on the remaining smaller subset of the initial training set. In each run of the GWOTLT, 50 evaluations of produced possible solutions are conducted, from which the best - the one with the highest fitness value is selected. To evaluate each solution, we train each solution for 10 epochs and then evaluate its performance. The selected solution is then trained for full 50 epochs on the whole given train dataset and finally evaluated on the given test set.

4.5 Evaluation method and metrics

Using the described experimental setup, we conducted two experiments, one using the CNN transfer learning approach without any optimization reported as *Baseline* and one utilizing the presented GWOTLT method reported as *GWOTLT*. For each of the experiments, we obtained six performance metrics: time - reported in seconds, AUC, $F - 1$ score, precision, and recall, reported in percents and kappa coefficient presented as a real value on the interval between 0 and 1.

To objectively evaluate the performance of the proposed method, we adapted the gold standard 10-fold cross-validation methodology, where a dataset is divided into train and test sets in a ratio 90:10. Using the images from 9 out of 10 folds for the training and performing the performance evaluation on the remaining one fold. In the same manner, we repeated the whole process in total 10 times, each time leaving different fold out for the performance evaluation. The reported values are presented as average values over 10 folds if not specified otherwise.

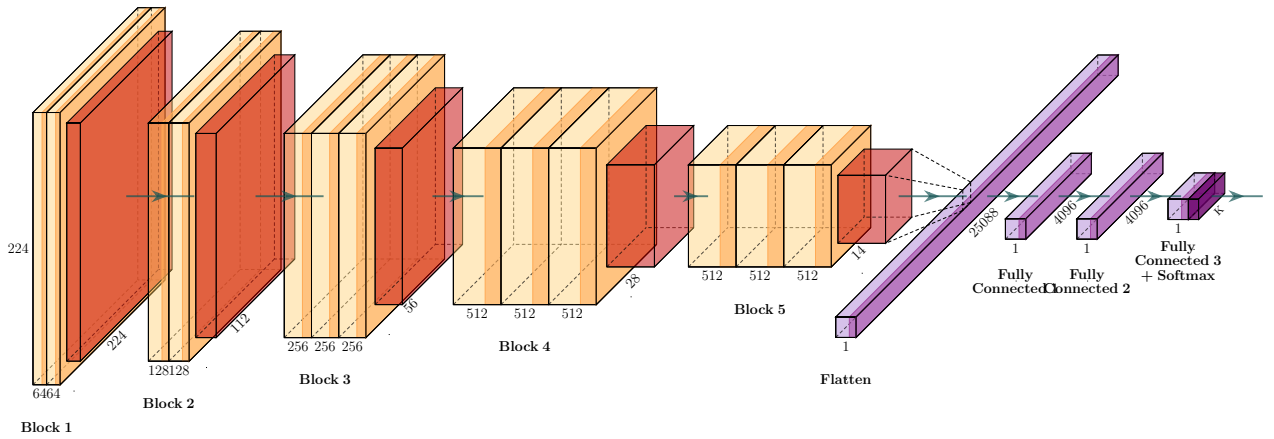


Figure 3: The architecture of the VGG16 convolutional neural network.

5. RESULTS

The obtained performance results from the conducted experiments are summarized in Table 3. Focusing on the time metrics, the reported results are expected, with the lowest time complexity being achieved by the *Baseline* method. On the other side, the proposed *GWOTLT* method is expected to have a higher time complexity in general due to the iterative nature of the proposed method. In our case, the *GWOTLT* method performed worse in the aspect of time complexity, roughly by a factor 15.

Analyzing presented classification performance metrics, the *GWOTLT* method is standing out with achieved best results on all of the reported performance metrics. The AUC, $F-1$, precision and recall metrics are higher by a margin of 4%, 5.18%, 2.27%, 7% respectively in comparison to the baseline method. Focusing on the kappa coefficient values, we can observe that the *GWOTLT* achieved a near-perfect agreement with kappa coefficient at 0.82 and outperformed the baseline method by a margin of 0.08. Looking at the standard deviations of the reported classification average metric values, we can observe that for all classification metrics, except for the precision, the best performing *GWOTLT* method is showing the smallest standard deviation. The greatest improvement of lowering the standard deviation the *GWOTLT* achieved for the recall metric by a margin of 10.38%, while the worst standard deviation is obtained for the precision metric where the *GWOTLT* lacks behind just by 0.99%.

6. CONCLUSIONS

In this paper, we presented the *GWOTLT* method which is a nature-inspired, population-based metaheuristics method for tuning the transfer learning approach of training the deep CNN. The *GWOTLT* method was implemented utilizing the GWO optimization algorithm and applied to the problem of identification of brain hemorrhage from the head CT scan images. The results obtained from the conducted experiments have proven that the proposed *GWOTLT* method seems to be very promising for the task of transfer learning tuning achieving higher classification performance for all of the measured classification metrics.

Metrics	<i>Baseline</i>	<i>GWOTLT</i>
Time [s]	49.10 ± 1.85	759.10 ± 59.67
AUC [%]	87.00 ± 9.19	91.00 ± 7.75
$F-1$ [%]	86.27 ± 11.03	91.45 ± 6.81
Precision [%]	88.62 ± 10.37	90.89 ± 11.36
Recall [%]	86.00 ± 17.13	93.00 ± 6.75
Kappa	0.74 ± 0.18	0.82 ± 0.15

Table 3: Comparison of average times, accuracies, AUCs, $F-1$ scores, precisions, recalls and kappa coefficients with standard deviations over 10-fold cross-validation.

In the future, we would like to expand our work to include various CNN architectures as a convolutional base for our *GWOTLT* method and also evaluate the performance of the proposed method against various medical imaging datasets.

Acknowledgments

The authors acknowledge the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0057).

7. REFERENCES

- [1] S. U. Akram, J. Kannala, L. Eklund, and J. Heikkilä. Cell segmentation proposal network for microscopy image analysis. In *Deep Learning and Data Labeling for Medical Applications*, pages 21–29. Springer, 2016.
- [2] E. Al Hadhrami, M. Al Mufti, B. Taha, and N. Werghi. Transfer learning with convolutional neural networks for moving target classification with micro-doppler radar spectrograms. In *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 148–154. IEEE, 2018.
- [3] U. Balasooriya and M. Perera. Intelligent brain hemorrhage diagnosis system. In *2011 IEEE International Symposium on IT in Medicine and Education*, volume 2, pages 366–370. IEEE, 2011.
- [4] P. Chang, E. Kuoy, J. Grinband, B. Weinberg, M. Thompson, R. Homo, J. Chen, H. Abcede,

- M. Shafie, L. Sugrue, et al. Hybrid 3d/2d convolutional neural network for hemorrhage evaluation on head ct. *American Journal of Neuroradiology*, 39(9):1609–1616, 2018.
- [5] F. Chollet et al. Keras, 2015.
- [6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [7] M. A. et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [8] Felipe Kitamura. Head CT - hemorrhage, 2018. Available at <https://www.kaggle.com/felipekitamura/head-ct-hemorrhage>, Accessed: 2019-02-21.
- [9] I. Fister Jr, X.-S. Yang, I. Fister, J. Brest, and D. Fister. A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*, 2013.
- [10] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *BioL Cybern.* 36 (1980) 193-202. *S. Shiotani et al./Neurocomputing 9 (1995) III-130*, 130, 1980.
- [11] M. Grewal, M. M. Srivastava, P. Kumar, and S. Varadarajan. Radnet: Radiologist level accuracy using deep learning for hemorrhage detection in ct scans. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 281–284. IEEE, 2018.
- [12] A. Helwan, G. El-Fakhri, H. Sasani, and D. Uzun Ozsahin. Deep networks in identifying ct brain hemorrhage. *Journal of Intelligent & Fuzzy Systems*, (Preprint):1–1, 2018.
- [13] M. Hussain, J. J. Bird, and D. R. Faria. A study on cnn transfer learning for image classification. In *UK Workshop on Computational Intelligence*, pages 191–202. Springer, 2018.
- [14] K. Jnawali, M. R. Arbabshirani, N. Rao, and A. A. Patel. Deep 3d convolution neural network for ct brain hemorrhage classification. In *Medical Imaging 2018: Computer-Aided Diagnosis*, volume 10575, page 105751C. International Society for Optics and Photonics, 2018.
- [15] A. Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.
- [16] D. Larsen-Freeman. Transfer of learning transformed. *Language Learning*, 63:107–129, 2013.
- [17] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [18] W. McKinney. Data structures for statistical computing in python. In S. van der Walt and J. Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.
- [19] S. Mirjalili, S. M. Mirjalili, and A. Lewis. Grey wolf optimizer. *Advances in engineering software*, 69:46–61, 2014.
- [20] C. Muro, R. Escobedo, L. Spector, and R. Coppinger. Wolf-pack (canis lupus) hunting strategies emerge from simple rules in computational simulations. *Behavioural processes*, 88(3):192–197, 2011.
- [21] K. Nogueira, O. A. Penatti, and J. A. dos Santos. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition*, 61:539–556, 2017.
- [22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [23] R. Rouhi, M. Jafari, S. Kasaei, and P. Keshavarzian. Benign and malignant breast tumors classification based on region growing and cnn segmentation. *Expert Systems with Applications*, 42(3):990–1002, 2015.
- [24] L. Shi, S. Xu, J. Zheng, J. Xu, and J. Zhang. Blood Pressure Management for Acute Intracerebral Hemorrhage: A Meta-Analysis. *Scientific Reports*, 7(1):14345, 2017.
- [25] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.
- [26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [28] S. Van Der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22, 2011.
- [29] G. Vrbančič, L. Brezočnik, U. Mlakar, D. Fister, and I. Fister Jr. NiaPy: Python microframework for building nature-inspired algorithms. *Journal of Open Source Software*, 3, 2018.
- [30] G. Vrbancic, I. J. Fister, and V. Podgorelec. Automatic Detection of Heartbeats in Heart Sound Signals Using Deep Convolutional Neural Networks. *Elektronika ir Elektrotehnika*, 25(3):71–76, jun 2019.
- [31] G. Vrbancic and V. Podgorelec. Automatic Classification of Motor Impairment Neural Disorders from EEG Signals Using Deep Convolutional Neural Networks. *Elektronika ir Elektrotehnika*, 24(4):3–7, aug 2018.
- [32] G. Vrbančič, I. Fister, Jr., and V. Podgorelec. Swarm intelligence approaches for parameter setting of deep learning neural network: Case study on phishing websites classification. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, WIMS '18*, pages 9:1–9:8, New York, NY, USA, 2018. ACM.