http://doi.org/10.35784/iapgos.5380 received: 31.07.2023 | revised: 01.10.2023 | accepted: 14.11.2023 | available online: 20.12.2023

USAGE OF ARTIFICIAL NEURAL NETWORKS IN THE DIAGNOSIS OF KNEE JOINT DISORDERS

Konrad Witkowski[1], Mikołaj Wieczorek[2]
[1]Lodz University of Technology, Lodz, Poland, [2]Synerise S.A., Krakow, Poland

Abstract. *Following article address the issue of automatic knee disorder diagnose with usage of neural networks. We proposed several hybrid neural net architectures which aim to successfully classify abnormality using MRI (magnetic resonance imaging) images acquired from publicly available dataset. To construct such combinations of models we used pretrained Alexnet, Resnet18 and Resnet34 downloaded from Torchvision. Experiments showed that for certain abnormalities our models can achieve up to 90% accuracy.*

Keywords: classification, MRI images, Resnet, Alexnet

ZASTOSOWANIE SZTUCZNYCH SIECI NEURONOWYCH W DIAGNOZIE SCHORZEŃ STAWU KOLANOWEGO

Streszczenie. *Niniejszy artykuł porusza temat automatycznej diagnozy uszkodzenia stawu kolanowego z zastosowaniem sieci neuronowych. Zaproponowano kilka hybrydowych sieci neuronowych, które podjęły próbę poprawnej klasyfikacji nieprawidłowości wykorzystując zdjęcia rezonansu magnetycznego pochodzące z publicznie dostępnego zbioru. Do konstrukcji kombinacji sieci skorzystano z pretrenowanych modeli (Alexnet, Resnet18, Resnet34) pobranych z Torchvision. Eksperyment pokazał, że dla klasyfikacji niektórych schorzeń modele osiągnęły nawet 90% skuteczności.*

Słowa kluczowe: klasyfikacja, zdjęcia MRI, Resnet, Alexnet

Introduction

Knee joint disorders are a problem strictly combined with the human aging process. Such disorders are outcomes of everyday work and accidents that lead to physical damage. One of the most effective diagnose methods of such injuries is the analysis of MRI images.

In this project we tried to construct a hybrid neural network architecture that could possibly accurately classify knee joint abnormalities using MRI images uploaded by Stanford University as "A Knee MRI Dataset And Competition" [8]. The researchers from Stanford ML Group also published an article [1] presenting results of their models which served as a reference point to scores achieved by our neural nets. We would also acknowledge the fact that for better understanding of our task we analysed Ahmed Besbes's implementation available here [5].

The goal of the whole project is to check whether a single person's exam consisting of 3 planes (axial, coronal and sagittal) indicate the occurrence of an injury like abnormality, ACL tears or meniscal tears. Each exam was viewed and tagged with labels by medical doctors.

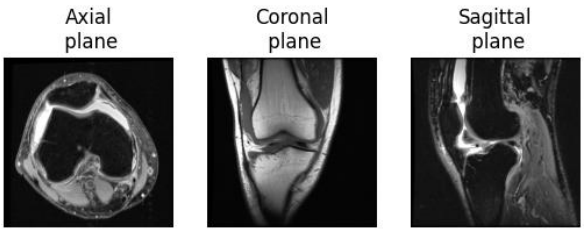

Fig. 1. Single exam's planes

This article is constructed as follows. At the beginning of the text we describe the dataset and the idea standing behind the experiment. After that we present the types of used neural nets and statistical methods. The last part of the paper is dedicated for experiment's results and summary.

1. Dataset characteristics

The dataset consists of 1 370 MRI examinations taken at Stanford University Medical Center. Each of the examination has 3 labels indicating presence of abnormality, ACL tears and meniscal tears. Occurrence of ACL tears or meniscal tears means that the abnormality label will be positive but it doesn't work the other way round. That means that the abnormality label covers not only ACL tears and meniscal tears but also other types of abnormalities not specified among labels.

MRI images were taken using various devices (GE Discovery, GE Healthcare, Waukesha, WI). Moreover two types of magnetic fields were used: 3.0 T (55.6% of exams) and 1.5 T for the rest of the exams.

Data uploaded by Stanford University was already pre-processed. That included converting DICOM (Digital Imaging and Communications in Medicine) files to png format and rescaling them to 256×256 resolution. Given that the images didn't have the same pixel intensity the researchers used standardization algorithm which based on pixel intensity taken from training dataset. The algorithm itself was run on both training and testing dataset.

In order to enhance the training dataset we performed augmentation consisting of random rotation, transposition and horizontal flip.

Table 1. Labels frequency

Label	Number of occurences
Abnormality	1 104
ACL Tears	319
Meniscal Tears	508

2. Experiment description

For each plane we attempted to build a single model (called submodel) which specialized in a specific label. To boost the performance of models we decided to use pretrained versions of Resnet18, Resnet34 and Alexnet downloaded from Torchvision [7] which served us as main parts of our submodels. Overview of their structures are available here: [2, 3]. The idea standing behind single net's functioning was to process the outcome of the pretrained model with average and max pooling, concatenate the results and finally perform calculations using fully connected layer. Whole structure is presented in Fig. 2.

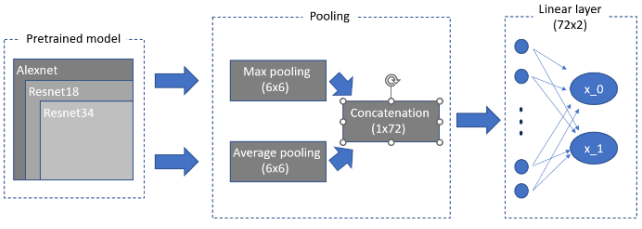

Fig. 2. Submodel's architecture scheme

The best suited model for classifying presence of specific label using given plane was chosen based on its performance and the results of McNemar's test run between all models.

The main models were composed out of 3 submodels each taking as input specific plane. Outcome of each submodel was sent to linear layer which ended up with 2 neurons.

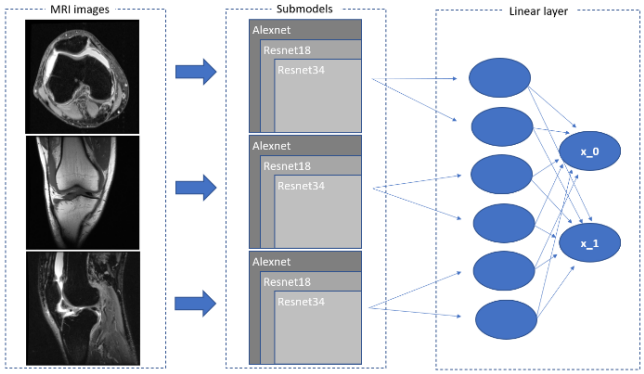

Fig. 3. Main models' architecture scheme

All neural nets were trained using binary cross entropy loss given by:

$$l = -w[y * ln\sigma(x) + (1-y) * \ln(1-\sigma(x)) \quad (1)$$

where x stands for predicted value, y for actual value, σ for logit function and w for weight loss.

To balance unequal label distribution we multiplied losses from actual positive observations with reversed proportion of number of actual positive observations to number of actual negative observations. This operation can be describes as follows:

$$w = \begin{cases} \frac{n.of\ actual\ negative\ obs.}{n.of\ actual\ positive\ obs.}, & y = 1 \\ 1, & y = 0 \end{cases} \quad (2)$$

3. Resnet

Resnet is a type of a neural network created by a group of researchers from Microsoft. Their main objective was solving the issue of degradation which takes place at the training process. The symptoms of this phenomenon appeared as deteriorated loss values not only on training set but also on test set whenever the construction of neural net was expanded with extra layers.

As a result, the researchers invented residual block which at that time differed from standard neural networks with the idea of using input vector at the beginning and at the end of set of layers.

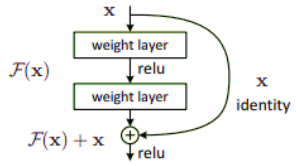

Fig. 4. Residual block scheme

Microsoft's researchers formulated a hypothesis saying that it's possible in an asymptotic way to estimate the outcomes of complicated functions. In the case of Resnet the authors took a step further and checked whether couple of connected layers are able to estimate outcome of residual function:

$$F(\boldsymbol{x}) = \mathrm{H}(\mathbf{x}) - \boldsymbol{x} \quad (3)$$

where H(x) is a covered mapping. It's possible to describe the way of working of residual block depicted in Fig. 4 in a following way:

$$\boldsymbol{y} = F(\mathbf{x}, \{W_i\}) + \boldsymbol{x} \quad (4)$$

where **x**, **y** are input and output tensor from the residual block. W_i describes the weights of *i*-th layer. The full version of *F* function can be expanded with 2 layers visible in Fig. 4 which gives:

$$F = W_2 \cdot \sigma(W_1 \boldsymbol{x}) \quad (5)$$

where σ is Relu activation function. The transformation presented in the equation (5) is called the skip connection and works only when dimensions of x and F are equal. When it's not the case then linear projection W_s is needed:

$$\boldsymbol{y} = F(\mathbf{x}, \{W_i\}) + \boldsymbol{x}W_s \quad (6)$$

Resnet's architecture is in fact a stack of residual blocks with implemented skip connections.

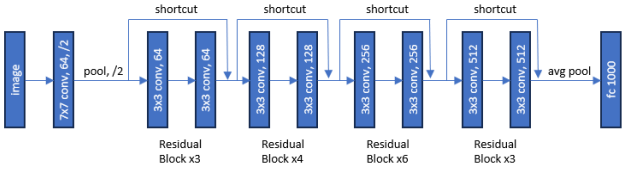

Fig. 5. Resnet34's architecture scheme

4. Alexnet

Alexnet is a type of neural network created by Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. Its success was based on several of innovations, whose Alexnet's creators were not always authors of, that were used all together in one architecture. Alexnet's training was spread around 2 graphic cards which allowed updating of 2 parallel mapping series and efficient memory management. Alexnet's structure begins with 3 convolutional layers which share data between graphic cards – input mapping for each of layer is structured from output tensors created by the previous layers placed on both graphic cards. Next 2 layers are convolutional type but in this case they are independent in sense of data sharing. They are followed by 3 fully connected layers.

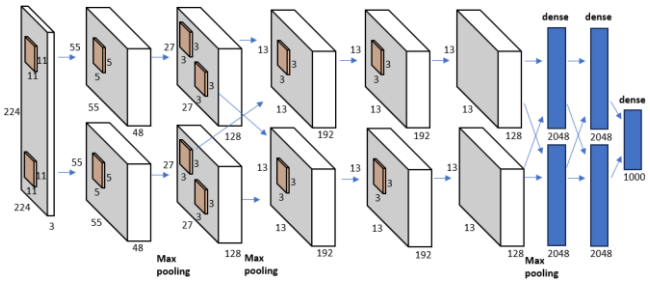

Fig. 6. Alexnet's architecture scheme

One of the most groundbreaking innovation was activation function called Relu which was described by following equation:

$$f(x) = \max(0, x) \quad (7)$$

At that time most of the neural networks used hyperbolic tangent which led to slower training tempo.

Another idea implemented in Alexnet was the local response normalization inspired by a phenomenon called the lateral inhibition which is characterized by excited neuron disabling its neighbors. This phenomenon leads to specialization of overexcited neurons in detecting certain patterns. Formally the local response normalization looks in the following way:

$$b^i_{x,y} = \frac{a^i_{x,y}}{(k + \alpha \sum_{j=\max(0,\ i-m/2)}^{\min(N-1,\ i+\frac{n}{2})} (a^j_{x,y})^2)^\beta} \quad (8)$$

where: $a^i_{x,y}$ is the outcome of applying *i*-th filter on element placed at (*x*,*y*) position and Relu function, $b^i_{x,y}$ is the normalized response for element placed at position (x,y) after applying *i*-th filter and *k*, *n*, *α*, *β* are the hyperparameters.

5. McNemar's test and Wilson's confidence interval

For comparison of models' classification results we performed McNemar's test whose overview is available here [6] and here [4]. This method uses convergence table which splits the same observations classified by two compared models into 4 groups presented in Table 2.

Table 2. Convergence table

	Classifier 1 correct results	Classifier 1 incorrect results
Classifier 1 correct results	a	b
Classifier 1 incorrect results	c	d

The null hypothesis states that both classifiers disagree to the same extend. If the null hypothesis is rejected, it means that there's a possibility that both classifiers disagree in a different way. To perform McNemar's test p value should be calculated which, depending on alfa value (here 0.05), confirms or rejects the null hypothesis:

p value > 0.05 → null hypothesis confirmed,

p value ≤ 0.05 → null hypothesis rejected.

In many cases chi square distribution is impossible to estimate since b + c < 25. Because of that reason we decided to use the exact p value given by the following equation:

$$exact\ value\ p = 2 \cdot \sum_{i=b}^{n} \binom{n}{i} \cdot 0.5^i \cdot (1-0.5)^{n-i} \quad (9)$$

where n = b + c.

In order to estimate the values of expected metrics on unseen before data we calculated Wilson confidence interval. A comprehensive overview is available here [9]. This method allowed us to construct range of expected results with given probability (95% in this project). Wilson confidence interval is given by the following equation:

$$p \approx (w^-, w^+) = \frac{1}{1+\frac{z^2}{n}} \cdot (\hat{p} + \frac{z^2}{2n}) \pm \frac{z}{1+\frac{z^2}{n}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}} \quad (10)$$

where n is the number of observations, z is z score for 95% confidence interval and $\hat{p}$ is the number of positive observations.

6. Results of submodels

For each label, for each plane, and for each pretrained model we performed a training process which lasted 10 epochs. Among 10 checkpoints we selected the one which obtained the highest accuracy on test set at the end of epoch. If at least 2 checkpoints achieved the same accuracy, then we chose the one which had the highest AUC (area under curve) result. At the end of the selection process we ended up with 27 submodels which we had to reduce to 9 – one submodel for (plane, label) pair.

To analyze submodel's performance we calculated the following statistics: accuracy, precision, recall, F1 score and AUC. To take a deeper look into delivered statistics we computed Wilson confidence interval. Furthermore we calculated p values using McNemar's test to find out whether submodels are statistically different.

Tables 3 and 4 present an example of results obtained by 3 submodels dedicated for abnormality classification using axial plane. In this case because of the high p values we decided to move forward with Resnet18 which has the least number of parameters.

Table 3. Classification of abnormality using axial plane

Model	Accuracy	Precission	Recall	F1	AUC
Alexnet	0.83 (0.753, 0.887)	0.86 (0.787, 0.911)	0.94 (0.882, 0.97)	0.9 (0.833, 0.942)	0.688 (0.601, 0.764)
Resnet18	**0.87 (0.798, 0.919)**	**0.87 (0.798, 0.919)**	**0.98 (0.936, 0.994)**	**0.92 (0.857, 0.956)**	**0.709 (0.623, 0.783)**
Resnet34	0.86 (0.787, 0.911)	0.86 (0.787, 0.911)	0.98 (0.936, 0.994)	0.92 (0.857, 0.956)	0.689 (0.602, 0.756)

Table 4. p values between submodels classifying abnormality using axial plane

Models		p value
Resnet18	Resnet34	1
Resnet18	Alexnet	0.424
Resnet34	Alexnet	0.648

We would also like to present insight into training process of chosen model. Fig. 7 and Fig. 8 show us that, even though all models were pretrained beforehand, the loss levels reached during training and testing looked different for submodel equipped with Alexnet and its equivalents with Resnets. The Alexnet submodel needed much more time to reach loss level represented by Resnet submodels.

Using the same strategy as described in the given example we selected the rest of submodels whose overview is presented in Tables 5 and 6.

Table 5. Final submodels chosen to build main models

Model	Accuracy	Precission	Recall	F1	AUC
Abnormality					
Resnet18 (axial)	0.87 (0.798, 0.919)	0.87 (0.798, 0.919)	0.98 (0.936, 0.994)	0.92 (0.857, 0.956)	0.709 (0.623, 0.783)
Resnet18 (coronal)	0.82 (0.742, 0.878)	0.85 (0.775, 0.903)	0.95 (0.895, 0.977)	0.9 (0.833, 0.942)	0.654 (0.565, 0.733)
Resnet18 (sagittal)	0.82 (0.742, 0.878)	0.82 (0.742, 0.878)	0.98 (0.936, 0.994)	0.89 (0.821, 0.934)	0.589 (0.5, 0.673)
ACL tears					
Resnet18 (axial)	0.75 (0.666, 0.819)	0.85 (0.775, 0.903)	0.54 (0.451, 0.627)	0.66 (0.571, 0.739)	0.731 (0.645, 0.802)
Resnet34 (coronal)	0.85 (0.775, 0.903)	0.86 (0.787, 0.911)	0.8 (0.72, 0.862)	0.83 (0.753, 0.887)	0.845 (0.77, 0.899)
Resnet34 (sagittal)	0.83 (0.753, 0.887)	0.83 (0.753, 0.887)	0.8 (0.72, 0.862)	0.81 (0.731, 0.87)	0.83 (0.753, 0.887)

Table 6. Final submodels chosen to build main models

Model	Accuracy	Precission	Recall	F1	AUC
Meniscal tears					
Resnet18 (axial)	0.67 (0.582, 0.748)	0.62 (0.531, 0.702)	0.62 (0.531, 0.702)	0.62 (0.531, 0.702)	0.661 (0.572, 0.739)
Resnet18 (coronal)	0.78 (0.698, 0.845)	0.75 (0.666, 0.819)	0.73 (0.644, 0.801)	0.74 (0.655, 0.81)	0.77 (0.687, 0.836)
Resnet34 (sagittal)	0.72 (0.634, 0.793)	0.72 (0.634, 0.793)	0.6 (0.511, 0.683)	0.65 (0.561, 0.729)	0.71 (0.623, 0.784)

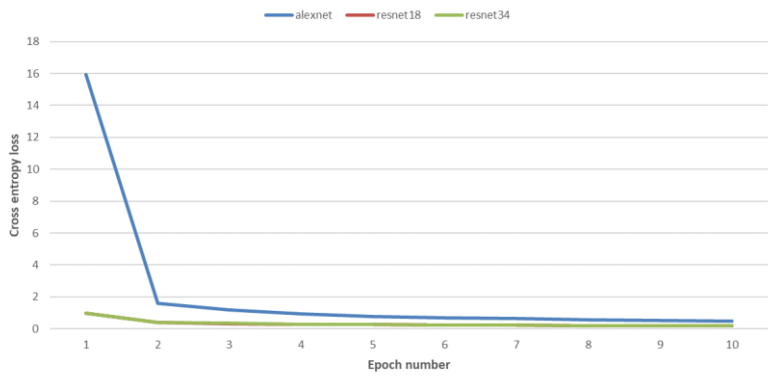

Fig. 7. Loss levels of submodels (classifying abnormality on train dataset) at the end of each epoch

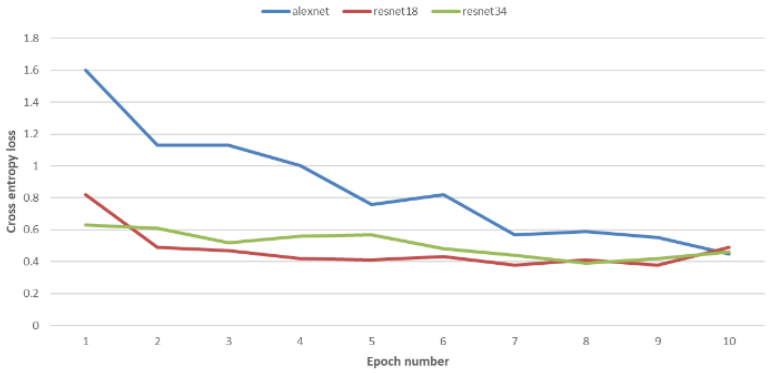

Fig. 8. Loss levels of submodels (classifying abnormality on test dataset) at the end of each epoch

7. Results of main models

To assess main models' effectiveness we decided to calculate the same metrics as in the submodels' cases but this time we expanded the analysis with specificity.

Table 7. Final models comparison compared with equivalents from Stanford University

Model	Accuracy	Precission	Specificity	Recall	F1	AUC
Abnormality						
Stanford's model	0.85 (0.775, 0.903)	No data	0.88 (0.800, 0.929)	0.71 (0.500, 0.862)	No data	0.94 (0.895, 0.937)
Authorial model	0.87 (0.798, 0.919)	0.86 (0.787, 0.911)	0.99 (0.952, 0.998)	0.4 (0.317, 0.489)	0.92 (0.857, 0.956)	0.69 (0.607, 0.77)
ACL tears						
Stanford's model	0.87 (0.794, 0.916)	No data	0.759 (0.635, 0.850)	0.97 (0.890, 0.991)	No data	0.97 (0.938, 0.965)
Authorial model	0.9 (0.833, 0.942)	0.9 (0.833, 0.942)	0.87 (0.798, 0.919)	0.924 (0.862, 0.959)	0.88 (0.81, 0.927)	0.9 (0.83, 0.94)
Meniscal tears						
Stanford's model	0.725 (0.639, 0.797)	No data	0.710 (0.587, 0.808)	0.74 (0.616, 0.837)	No data	0.85 (0.78, 0.847)
Authorial model	0.77 (0.687, 0.836)	0.69 (0.602, 0.766)	0.85 (0.775, 0.903)	0.706 (0.619, 0.78)	0.76 (0.676, 0.828)	0.78 (0.694, 0.841)

In each of 3 pairs of compared models the same pattern can be spotted – the differences in accuracy between Stanford's models and authorial models are low. For instance accuracy of Stanford's model responsible for classifying images with abnormality presence reached 0.85 in comparison to authorial model's 0.87.

It seems that the neural nets created by Stanford ML Group work much better in detecting those images which don't have sought label. In contrast to them authorial models excel in finding disorders in the images that actually present joint with disorder. For example recall and specificity of Stanford's model classifying ACL tears reached levels of 0.759 and 0.924. The same metrics for authorial model stood at 0.9 and 0.924.

It's worth mentioning that Stanford's neural networks overtook the authorial models when it comes to the AUC levels. The possible explanation for that could be the difference in a way of selecting submodel's checkpoint among epochs during training. The original Stanford's paper mentions that the researched chose those versions which had the lowest averaged loss counted within epoch. On the other hand authorial submodels were picked according to the highest accuracy.

8. Summary

In conclusion we would like to say that the created models that served to classify 3 types of knee joint disorder achieved comparable results as their equivalents from Stanford University. Their differences in a way of selection are with no doubts a good material for further research.

It seems that the topic of classifying knee joint injuries using neural nets is worth spending much more time on it. In our opinion aspects like choice of pretrained model or the construction of submodel could be much better explored. It's also clear to us that such models should help the medical doctors not only in the proper classification but also In pointing the place where injury is located. This was the main idea of Researchers from Stanford University who implemented class activation mapping – a heatmap generating technique which shows which part of the image were significant in classification.

References

[1] Bien N. et al.: Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation of MRNet. PLoS Med 15(11), 2018, e1002699 [http://doi.org/10.1371/journal.pmed.1002699].
[2] He K., Zhang X., Ren S., Sun J.: Deep Residual Learning for Image Recognition. Computer Vision and Pattern Recognition 2015, arXiv:1512.03385.
[3] Krizhevsky A., Sutskever I., Hinton G. E.: ImageNet Classification with Deep Convolutional Neural Networks. F. Pereira, C. J. Burges, L. Bottou and K. Q. Weinberger: Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012.
[4] https://en.wikipedia.org/wiki/McNemar%27s_test
[5] https://github.com/ahmedbesbes/mrnet
[6] https://machinelearningmastery.com/mcnemars-test-for-machine-learning/
[7] https://pytorch.org/vision/stable/models.html
[8] https://stanfordmlgroup.github.io/competitions/mrnet/
[9] https://www.mikulskibartosz.name/wilson-score-in-python-example/

M.Sc. Konrad Witkowski
e-mail: k.l.p.witkowski@gmail.com

Konrad Witkowski received his M.Sc. from the SGH Warsaw School of Economics. His research interest include neural nets, machine learning etc.

http://orcid.org/0009-0004-2916-8672

M.Sc. Mikołaj Wieczorek
e-mail: mwieczorek.doc@gmail.com

Mikołaj Wieczorek, holds a Master's in Operational Research with Data Science from the University of Edinburgh. Professionally works as ML/MLOps Engineer. His expertise lies in cloud infrastructure, AI model deployment and optimisation. His research interest include Computer Vision, NLP and recommender systems.

http://orcid.org/0000-0003-4397-3331