1 Introduction

Deep learning, introduced by Yann Lecun [1], combines multiple Machine Learning techniques and allows computational models that are composed of numerous processing layers to learn representations of data with various levels of abstraction. These methods have dramatically improved the state-of-the-art in many domains, such as image classification or speech recognition. Deep learning discovers complex structures in large datasets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the features in the previous layer.

Convolutional neural networks (ConvNets) are designed to process data that comes in the form of multiple arrays. ConvNets are suited for images, video, speech and audio processing, and they have been used extensively in the last years for automatic classification tasks [2][3][4]. On image classification tasks, each colour channel is represented by a 2D array. In this case, convolutional layers extract the main features from the pixels of the images and, after that, a fully connected layer classify every sample based on its extracted features. At the output of the convolutional net, a softmax function provides the probabilities of the set of classes predefined in the model for classification tasks. These outputs are compared against the correct values.

The backpropagation algorithm is a stochastic gradient descent algorithm that minimises a predefined loss function. It has been used in the last years in many works [5][6][7][8] for training swallow and deep neural networks. It updates the layer's parameters after backpropagating the loss function gradients through the network. Learning rate hyper-parameter controls the strength of the changes that are applied to those parameters. Some works have checked the importance of finding the optimal value for this parameter [9] and have tried different approaches to try to improve the training process [10].

Batch normalization is another technique that is used for this kind of networks. It reduces the internal covariate shift by normalizing layer inputs. It was presented in 2015 [11] and gives a critical enhancement to the training phase. Batch size is an important hyper-parameter that should be adjusted as it affects the layer's parameters updates and also the normalization [12][13].

Ordinal classification problems are those where labels are ordered. Multiple metrics exist in the literature of machine learning and statistics [14][15]. Kappa index is a well-known statistic coefficient defined by Cohen [16] to measure inter-rater agreement on classifying elements into a set of categories. Later, Weighted Kappa (WK) is a modified version of the Kappa statistic calculated to allow assigning different weights to different levels of aggregation between two variables. Weighted Kappa loss was described in previous work [8] and is a cost function that is based on the WK metric. These functions are indicated for problems where different inter-classes weights are assigned, and that work proved that using these functions improves the model performance and reduce the overfitting risk. This kind of problems is associated with ordinal data processing. Those weights are predefined and depend on the type of the chosen WK. In a linear penalization, the weight is proportional to the distance between the predicted and the real class. In the Quadratic Weighted Kappa (QWK) the penalization is proportional to the square of the distance. In this work, we will combine the QWK metric and the QWK loss.

Softmax function is widely used for the output layer in neural networks for classification tasks. It is a simple and efficient function for multi-class problems but not the best when working with ordinal data. In this paper, the Proportional Odds Model (POM) [17] will be used instead of the softmax function. Different link functions will be explored to compare their performance. Also, the influence of other parameters like the learning rate of the optimization algorithm and the batch size will be studied. We will contrast the results obtained with statistical analysis to provide more robust results. An approximated ANOVA III test will be performed because of the limitations of the computational time required to run a higher number of executions.

The experiments will be run using two different ordinal datasets: Diabetic Retinopathy, that contains high-resolution fundus images related with diabetes disease, and Adience, that is formed of human faces images that are assigned an age range.

This paper is organised as follows: in Section 2, we take a look at previous works related to this paper, in Section 3 we describe the model, the experiments and the datasets used, in Section 4 we present the results obtained and the statistical analysis and finally in Section 5 we expose the conclusions of this study.

2 Related work

J. de la Torre et al. [8] proposed the use of QWK loss function for the optimization algorithm. They compared this cost function against the traditional log-loss function across three different databases, including the Diabetic Retinopathy database as the most complex one. They proved that their function could improve the results as it reduces the overfitting and the required training time. Also, they checked the importance of hyper-parameter tuning.

Christopher Beckham and Christopher Pal [18] proposed a straightforward technique to constrain discrete ordinal probability distributions to be unimodal via the use of the Poisson and binomial probability distributions. They evaluated this approach in the context of deep learning on two large ordinal image datasets, including the Adience dataset used in this paper, and they obtained promising results.

The Exponential Linear Unit (ELU), used in this paper, was described in previous work from D.-A. Clevert et al. [19]. They proved that ELUs alleviate the vanishing gradient problem via the identity for positive values. In their experiments, ELUs lead not only to faster learning but also to significantly better generalization performance than ReLUs and LReLUs on networks with more than five layers.

Sergey Ioffe and Christian Szegedy [11] described Batch Normalization and its benefits in previous work. It reduces the internal covariate shift by normalizing layer inputs. Their method draws its strength from making normalization a part of the model architecture and performing the normalization for each training batch. It allows them to use higher learning rates and be less careful about initialization. It also eliminates the need for using regularization techniques like Dropout.

3 Experiments

3.1 Data

We make use of two ordinal datasets appropriate for deep neural networks:

- *Diabetic Retinopathy (DR)*[1]. DR is a dataset consisting of extremely high-resolution fundus image data. The training set consists of 17563 pairs of images (where a pair consists of a left and right eye image corresponding to a patient). In this dataset, we try and predict from five levels of diabetic retinopathy: no DR (25810 images), mild DR (2443 images), moderate DR (5292 images), severe DR (873 images), or proliferative DR (708 images). The images are taken in variable conditions: by different cameras, illumination conditions and resolutions. These images come from the EyePACS dataset that was used in a Diabetic Retinopathy Detection competition that was hosted on the Kaggle platform. Also, this dataset was used in later works [8][20] and ordinal techniques (such as an ordinal cost function) were applied in order to achieve better performance. A validation set is set aside, consisting of 10% of the patients in the training set. The images are resized to 128 by 128 pixels. Data augmentation techniques are applied to achieve a higher number of samples.

- *Adience*[2]. This dataset consists of 26580 faces belonging to 2284 subjects. We use the form of the dataset where faces have been pre-cropped and aligned. The dataset was preprocessed, using the methods described in a previous work [18], so that the images are 256 pixels in width and height, and pixels values follow a $(0; 1)$ normal distribution. The original dataset is split into five cross-validation folds. The training set consists of merging the first four folds which comprise a total of 15554 images. From this, 10% of the images are held out as part of a validation set. The last fold is used as test set.

3.2 The model

A convolutional neural network (CNN) has been used for both datasets. The architecture of this CNN is presented in the Table 1.

Every convolutional layer is followed by an ELU activation layer [19] and a batch normalization [11]. At the output, a Proportional Odds Model is used with different link functions [17]. The logit link function is commonly used within POM. In this paper, we are comparing other link functions like probit or complementary log-log with the logit link. These three types of links are explained below.

[1] https://www.kaggle.com/c/diabetic-retinopathy-detection/data

[2] http://www.openu.ac.il/home/hassner/Adience/data.html

Layer	Output shape
Conv_32_3x3	254x254x32
Conv_32_3x3	252x252x32
MaxPool_2x2	126x126x32
Conv_64_3x3	124x124x64
Conv_64_3x3	122x122x64
MaxPool_2x2	61x61x64
Conv_128_3x3	59x59x128
Conv_128_3x3	57x57x128
MaxPool_2x2	28x28x128
Conv_128_3x3	26x26x128
Conv_128_3x3	24x24x128
MaxPool_2x2	12x12x128
Conv_128_4x4	9x9x128
Dense_1_output	1

Table 1: Description of the architecture used in the experiments. For convolutional layers, Conv_N_WxH, where N is the number of filters, W the filter width and H the filter height. Stride is 1 for every convolutional layer. For max pool layers, MaxPool_SxS, where S is the pool size.

- *Logit.* Logit link function is the most widely used function for Proportional Odds Models. These kind of models are also called Cummulative Logit Models. The logit link is shown in Eq. 1.

$$logit[P(Y_i \leq j)] = \alpha_j + \beta' x_i, \quad j = 1, ..., c-1 \quad (1)$$

- *Probit.* Probit link function is the inverse of the standard normal cummulative distribution function (cdf). Its expression is shown in Eq. 2.

$$\Phi^{-1}[P(Y \leq j)] = \alpha_j + \beta' x, \quad j = 1, ..., c-1 \quad (2)$$

- *Complementary log-log.* Unlike logit and probit, complementary log-log function is not symmetric. With a continuous predictor x, for example, $P(Y \leq j)$ approaches 0 at a different rate than it approaches 1. Complementary log-log expression is shown in Eq. 3.

$$log[-log[1 - P(Y \leq j)]] = \alpha_j + \beta' x, \quad j = 1, ..., c-1 \quad (3)$$

3.3 Procedure

The model is optimized using a batch based first-order optimization algorithm called Adam [21]. We study different initial learning rates in order to find the optimal one for each problem. We apply an exponential decay across training epochs to the initial learning rate.

Quadratic Weighted Kappa Loss is considered as loss function for this optimizer as it gives better performance for ordinal classification problems.

Both datasets have been artificially equalised using data augmentation techniques [22][7] based on image cropping and zooming, horizontal and vertical flipping, brightness adjustment and random rotations. In the case of Diabetic Retinopathy Detection, the epoch size has been fixed to 100000 images per epoch. For the Adience dataset, epoch size is the number of images in the training set.

The model is evaluated with Quadratic Weighted Kappa metric (QWK) [23]. This evaluation measure gives a higher weight to the errors that are further from the correct class.

Experiments were run with the standard cross-entropy loss and the softmax function too in order to prove the performance improvement of the QWK loss and the Proportional Odds Model. The results of these experiments are analysed in Section 4.4.

3.4 Parameters

Three different parameters have been considered: learning rate, batch size and link function for the final output layer.

- *Learning rate.* Learning rate is one of the most critical hyper-parameters to tune for training deep neural networks. Optimal learning rate can vary depending on the dataset and the CNN architecture. Previous works have presented some techniques that adjust this parameter in order to achieve better performance [10][9]. Within this work, we have considered three different values for this parameter: 10^{-2}, 10^{-3} and 10^{-4}.

- *Batch size.* Batch size is also an important parameter as it controls the number of weight updates that are made on every epoch. It can affect the training time and the model performance. In this paper, we have tried three separate batch sizes: 5, 10 and 15.

- *Link function.* Different link functions have been used for the POM at the last layer output: logit, probit and complementary log-log.

4 Results

In this section, we present the results of the experiments. For each dataset, we show a table with the detailed experiments done training the model with each combination of parameters. Every parameter combination has been run five times. These tables show the average quadratic weighted kappa across these five executions for validation and test values.

4.1 Diabetic Retinopathy

Detailed results for the Diabetic Retinopathy dataset are presented in Table 2.

BS	LR	LF	κ_{val}	κ_{test}
5	10^{-2}	Logit	0.44888	0.4163
		Probit	0.46724	0.45972
		c log-log	0.42854	0.41448
	10^{-3}	Logit	0.56496	0.55356
		Probit	0.57084	0.56392
		c log-log	0.54796	0.53436
	10^{-4}	Logit	0.52884	0.52032
		Probit	0.53802	0.52302
		c log-log	0.53824	0.51974
10	10^{-2}	Logit	0.5497	0.53142
		Probit	0.52124	0.50806
		c log-log	0.4276	0.4227
	10^{-3}	Logit	0.58036	0.57686
		Probit	0.56536	0.5581
		c log-log	**0.5883**	**0.5815**
	10^{-4}	Logit	0.53158	0.5385
		Probit	0.5456	0.54082
		c log-log	0.53928	0.53742
15	10^{-2}	Logit	0.55828	0.55076
		Probit	0.54112	0.5343
		c log-log	0.56676	0.56428
	10^{-3}	Logit	0.55748	0.55106
		Probit	0.58182	0.57894
		c log-log	0.5634	0.55938
	10^{-4}	Logit	0.54686	0.54342
		Probit	0.53258	0.53278
		c log-log	0.539	0.53828

Table 2: Diabetic Retinopathy results. BS stands for Batch Size, LR for Learning Rate and LF for link function.

The best mean WK value was obtained with the complementary log-log link function using a batch size of 10 and a learning rate of 10^{-3}.

4.2 Adience

Results for the experiments made with the Adience dataset are shown in Table 3.

BS	LR	LF	κ_{val}	κ_{test}
5	10^{-2}	Logit	0.72072	0.68408
		Probit	0.77608	0.72574
		c log-log	0.7543	0.70742
	10^{-3}	Logit	0.7146	0.67868
		Probit	0.81906	0.77154
		c log-log	0.57196	0.55608
	10^{-4}	Logit	0.54432	0.51934
		Probit	0.5436	0.52422
		c log-log	0.54802	0.52484
10	10^{-2}	Logit	0.78872	0.72994
		Probit	0.79332	0.7475
		c log-log	0.78004	0.7326
	10^{-3}	Logit	0.49686	0.47112
		Probit	0.59778	0.58476
		c log-log	0.52016	0.49968
	10^{-4}	Logit	0.60972	0.56814
		Probit	0.54638	0.52632
		c log-log	0.75428	0.52094
15	10^{-2}	Logit	0.79104	0.74388
		Probit	**0.84254**	**0.78666**
		c log-log	0.78802	0.74624
	10^{-3}	Logit	0.56946	0.52978
		Probit	0.70262	0.65416
		c log-log	0.49242	0.4732
	10^{-4}	Logit	0.55202	0.52484
		Probit	0.54288	0.5174
		c log-log	0.53826	0.52542

Table 3: Adience results. BS stands for Batch Size, LR for Learning Rate and LF for link function.

The best mean WK value was obtained with the probit link function using a batch size of 15 and a learning rate of 10^{-2}.

4.3 Statistical analysis

In this subsection, a statistical analysis will be performed in order to obtain robust conclusions from the results presented in this section.

The significance and relative importance of the parameters concerning the results obtained, as well as suitable values for each, were obtained using the ANalysis Of the VAriance (ANOVA) test.

The ANOVA test [24] is one of the most widely used statistical techniques. ANOVA is essentially a method of analysing the variance to which a response is subject into its various components, corresponding to the sources of variation which can be identified.

ANOVA, in this case, examines the effects of three quantitative variables (termed factors) on one quantitative response. Considered factors are the link function, which can be logit, probit and complementary log-log, the learning rate for the Adam optimization algorithm (10^{-2}, 10^{-3} and 10^{-4}) and the batch size

(5, 10 and 15).

Following the setup of the previous study, we performed an ANOVA III analysis and multiple comparison tests. We assume that five executions are enough to do the statistical tests because of the computational time limitations.

The test shows that there is no batch size whose results are significatively better than the results of all other batch sizes. This does not mean that these differences could not exist for specific numbers of samples per batch. So, in order to determine for each type of function whether a batch size is better than the others, we have performed an ANOVA I analysis - where the only factor is the batch size - and multiple comparison tests.

We denote by $W_{i,j,k,l}(i = 1, ..., 3; j = 1, ..., 3; k = 1, ..., 3)$ the value observed when the first factor is at the i-th level, the second at the j-th level and the third at k-th level. We assume that the two factors do not act independently and therefore there exists an interaction between them. In this case, the observations fit Eq. 4.

$$\begin{aligned} W_{i,j,k,l} =& \mu + L_i + P_j + B_k + LP_{i,j} + LB_{i,k} \\ &+ PB_{j,k} + LPB_{i,j,k} + \epsilon_{i,j,k,l} \end{aligned} \quad (4)$$

where μ is the fixed effect that is common to all the populations; L_i is the effect associated with the *i-th* level of the link factor (logit, probit, complementary log-log); P_j is the effect associated with the *j-th* level of the parameter factor (10^{-4}, 10^{-3}, 10^{-2}) and B_k is the effect associated with the *k-th* level of the batch size factor (5, 10, 15). The term $LP_{i,j}$ denotes the joint effect of the presence of level i of the first factor and level j of the second one; this, therefore, is denominated the interaction term between L and P factors. The same interaction effect is appreciated on $LB_{i,k}$, $PB_{j,k}$ and $LPB_{i,j,k}$. The term $\epsilon_{i,j,k,l}$ is the influence on the result of everything that could not be assigned or of random factors. $W_{i,j,k,l}$ is the quadratic weighted kappa measure, the response variable used to perform the statistical analysis.

We consider some hypotheses testing where the null hypothesis is proposed that each term of the above equation is independent of the levels involved. The hypotheses for the levels of the L factor are

$$H_0 \equiv L_1 = L_2 = L_3$$

$$H_1 \equiv \text{ some } L_i \text{ is different}$$

The same hypotheses are made for the other factors. In this way, we test in the null hypothesis that all of the population means are equal against an alternative hypothesis that there is at least one mean that is not equal to the others.

The hypothesis associated with the interaction between L and P is

$$H_0 \equiv LP_{i,j} = 0 \quad \forall i, j$$

$$H_1 \equiv \exists LP_{i,j} \neq 0$$

Similar hypotheses can be assumed for the interaction between the other factors.

The analysis of variance table containing the sum of squares, degrees of freedom, mean square, test statistics and significance level represents the initial study in a compact form.

Response variable WK

Source	S.S.	D.F.	M.S.	F-ratio	Sig.
Model	37.860	9	4.207	1562.840	.000
L factor	.057	2	.029	10.646	.000
P factor	.121	2	.060	22.468	.000
LP factors	.057	4	.014	5.261	.001
Error	.339	126	.003		
Total	38.199	135			

Table 4: ANOVA III for the analysis of the main factors in the design of a Convolutional Ordinal Neural Network for the Retinopathy dataset.

The results of the ANOVA III test for the Diabetic Retinopathy dataset are resumed in Table 4. There are significative differences in the means depending on the link function and also depending on the learning rate for $\alpha = 0.05$. Moreover, an interaction between the link function and the learning rate can be recognised (p-value = 0.001).

As Table 4 showed that there exist significative differences between the means, we are now analysing those differences. A post-hoc multiple comparison test has been performed on the average WK obtained. An HSD Tukey's test was made under the null hypothesis that the variance of the error of the dependent variable is the same between groups. The best link function is the complementary log-log, though it doesn't have significative differences with the probit function ($\alpha = 0.05$). There are differences between the complementary log-log and the probit function concerning the logit function.

The results have been obtained following a similar methodology with the different values of the learning

parameter including the HSD Tukey's test and the learning parameter ranking. Significative differences can be observed in the mean values for $\eta = 10^{-3}$ with respect to the other values of the parameter ($\alpha = 0.05$). Also, there are differences for $\eta = 10^{-4}$ concerning $\eta = 10^{-2}$.

Response variable WK

Source	S.S.	D.F.	M.S.	F-ratio	Sig.
Model	52.328	15	3.489	1426.170	.000
L factor	.027	2	.014	5.582	.005
P factor	1.031	2	.516	210.809	.000
B factor	.089	2	.045	18.253	.000
LP factors	.183	4	.046	18.695	.000
PB factors	.124	4	.031	12.695	.000
Error	.294	120	.002		
Total	52.622	135	3.489		

Table 5: ANOVA III for the analysis of the main factors in the design of a Convolutional Ordinal Neural Network for the Adience dataset.

The results of the ANOVA III test for the Adience dataset are shown in Table 5. The interactions where there are not significative differences have been omitted. First, the factor associated with the link function is analysed. The differences between the means are significative (p-value = 0.005). Secondly, there are significative differences concerning the learning rate factor too (p-value = 0.000). Also, there are differences for the means for the batch size factor. Significative interactions exist between two pairs of factors: link function and learning rate, and learning rate and batch size (p-value = 0.000 for both cases).

A post-hoc multiple comparison test has been performed on the average WK obtained for this dataset too. Under the null hypothesis that the variance of the error of the dependent variable is the same between groups, an HSD Tukey's test has been done. The results showed that the best link function is the logit one, in this case, with 0.632438 as mean WK value. However, there are not significative differences between this function and the complementary log-log (p-value = 0.110). It has differences with the probit function though (p-value = 0.003). The complementary log-log link reported the best results after the logit function, having a mean WK of 0.611287. However, it doesn't show significative differences with the probit function. The results have been obtained following a similar methodology with the different values of the learning parameter including the HSD Tukey's test and the learning parameter ranking. The test reported significative differences between $\eta = 10^{-2}$ (mean value 0.733784) and the rest of values of this parameter ($\alpha = 0.005$), being this one the best value for this factor. The best value after 10^{-2} is 10^{-3}, which have a mean WK value of 0.579889. Lastly, the best value for the batch size parameter is 10 (mean value 0.648700) as it has significative differences with the rest of values. The second best value is 5 (mean 0.605533), but it has no significative differences with 15 (p-value = 0.194).

4.4 Statistical comparison between nominal and ordinal methods

The same experiments described in Section 3 were repeated using the cross-entropy instead of the QWK as loss function for the optimizer and the softmax function instead of the Proportional Odds Model for the output of the network. The results for both datasets are shown in Table 6.

Dataset	BS	LR	κ_{val}	κ_{test}
Retinopathy	5	10^{-2}	0.2210	0.2050
Retinopathy	5	10^{-3}	0.2807	0.2654
Retinopathy	5	10^{-4}	0.4580	0.4449
Retinopathy	10	10^{-2}	0.2972	0.2997
Retinopathy	10	10^{-3}	0.3972	0.3989
Retinopathy	10	10^{-4}	**0.4972**	**0.4967**
Retinopathy	15	10^{-2}	0.3692	0.3676
Retinopathy	15	10^{-3}	0.4107	0.4162
Retinopathy	15	10^{-4}	0.4929	0.4859
Adience	5	10^{-2}	0.0605	0.0769
Adience	5	10^{-3}	0.7994	0.7313
Adience	5	10^{-4}	0.8009	0.7383
Adience	10	10^{-2}	0.0559	0.0576
Adience	10	10^{-3}	0.8239	0.7436
Adience	10	10^{-4}	0.7938	0.7218
Adience	15	10^{-2}	0.0593	0.0713
Adience	15	10^{-3}	**0.8309**	**0.7575**
Adience	15	10^{-4}	0.7867	0.7129

Table 6: Nominal method results. BS stands for Batch Size and LR for Learning Rate.

As we did in Section 4.3, we must analyse the effect of the batch size (with values 5, 10 and 15) and the learning rate (10^{-2}, 10^{-3} and 10^{-4}) factors have over the WK metric. So, we make an ANOVA II analysis for each dataset.

In the Diabetic Retinopathy dataset, there are significative differences between the mean values of WK depending on the batch size and the initial value of the learning rate. Also, no significative interaction was detected between these factors.

After that, a post-hoc Tukey's test shows that the best value for the batch size factor is 15 followed by 10. However, there are no significative differences between them, but there are differences with size 5. Also, the best value for the learning rate is 10^{-4} and there are significative differences with the rest of values.

A similar study was made for the Adience dataset. There are not significative differences for the mean value of WK w.r.t. the batch size factor. Also, there are no interactions between the batch size and the learning rate. The best results were obtained with 10^{-3} for the learning rate followed by 10^{-4}, without significative differences between them.

Finally, the conclusions of this study are exposed below:

- There are no interactions between both factors.
- Batch sizes 10 and 15 obtained the best results.
- Learning rate 10^{-4} is the best choice for both datasets as it the best for Diabetic Retinopathy, with significative differences, and it is the second for Adience, without significative differences with the best.

5 Conclusions

In this section, the conclusions obtained from this work are exposed. The first thing that we have noticed is that the optimal values for the different parameters considered are problem-dependant.

The complementary log-log function has the best average performance across both datasets. It offers the best results in Diabetic Retinopathy dataset and the second best result in Adience dataset. These results provide an opportunity for exploring new generalised link functions.

The best value for the learning rate parameter for Diabetic Retinopathy dataset is $\eta = 10^{-3}$ while this value is the second best option for Adience dataset. It can be considered a good value for this parameter when training the model with new datasets.

Both datasets have obtained the best performance with batch size 10.

Also, the statistical tests reported that there are interactions between some parameters like the link function and the learning rate, for Diabetic Retinopathy, or those factors in addition to learning rate and batch size, for Adience dataset.

The proposed POM model has improved the performance of the deep network regarding Weighted Kappa compared to the model that uses the softmax function. This enhancement is more notable for the Diabetic Retinopathy dataset. Also, it reduces the chance that the model gets stuck when training with some parameter configurations. So, the most significant improvements of these link functions are the WK performance increase, the reduction of the number of parameters configurations that should be tried to find the best one and the prevention of the over-fitting and the stagnation.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, p. 436, 2015.

[2] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution", in *European conference on computer vision*, Springer, 2014, pp. 184–199.

[3] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3476–3483.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[5] J. Leonard and M. Kramer, "Improvement of the backpropagation algorithm for training neural networks", *Computers & Chemical Engineering*, vol. 14, no. 3, pp. 337–341, 1990.

[6] X.-H. Yu, G.-A. Chen, and S.-X. Cheng, "Dynamic learning rate optimization of the backpropagation algorithm", *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 669–677, 1995.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] J. de la Torre, D. Puig, and A. Valls, "Weighted kappa loss function for multi-class classification of ordinal data in deep learning", *Pattern Recognition Letters*, vol. 105, pp. 144–154, 2018.

[9] A. Senior, G. Heigold, K. Yang, *et al.*, "An empirical study of learning rates in deep neural networks for speech recognition", in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, IEEE, 2013, pp. 6724–6728.

[10] L. N. Smith, "Cyclical learning rates for training neural networks", in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, IEEE, 2017, pp. 464–472.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *ArXiv preprint arXiv:1502.03167*, 2015.

[12] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima", *ArXiv preprint arXiv:1609.04836*, 2016.

[13] P. M. Radiuk, "Impact of training set batch size on the performance of convolutional neural networks for diverse datasets", *Information Technology and Management Science*, vol. 20, no. 1, pp. 20–24, 2017.

[14] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P. A. Gutiérrez, "Metrics to guide a multi-objective evolutionary algorithm for ordinal classification", *Neurocomputing*, vol. 135, pp. 21–31, 2014.

[15] N. Mehdiyev, D. Enke, P. Fettke, and P. Loos, "Evaluating forecasting methods by considering different accuracy measures", *Procedia Computer Science*, vol. 95, pp. 264–271, 2016.

[16] J. Cohen, "A coefficient of agreement for nominal scales", *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[17] A. Agresti, *Analysis of ordinal categorical data*. John Wiley & Sons, 2010, vol. 656.

[18] C. Beckham and C. Pal, "Unimodal probability distributions for deep ordinal classification", *ArXiv preprint arXiv:1705.05278*, 2017.

[19] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)", *ArXiv preprint arXiv:1511.07289*, 2015.

[20] À Nebot *et al.*, "Diabetic retinopathy detection through image analysis using deep convolutional neural networks", in *Artificial Intelligence Research and Development: Proceedings of the 19th International Conference of the Catalan Association for Artificial Intelligence, Barcelona, Catalonia, Spain, October 19-21, 2016*, IOS press, 2016, p. 58.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *ArXiv preprint arXiv:1412.6980*, 2014.

[22] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation", *Journal of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, 2001.

[23] A. Ben-David, "Comparison of classification accuracy using cohen's weighted kappa", *Expert Systems with Applications*, vol. 34, no. 2, pp. 825–832, 2008.

[24] R. G. Miller Jr, *Beyond ANOVA: Basics of applied statistics*. Chapman and Hall/CRC, 1997.