Classification of a Small Imbalanced Dataset of Vine Leaves Images using Deep Learning Techniques

Amjad Balawi, Abdullah Al Zoabi, José Luis Seixas Junior, and Tomáš Horváth

Department of Data Science and Engineering
ELTE – Eötvös Loránd University
http://t-labs.elte.hu/
Faculty of Informatics, 3in Research Group, Martonvásár, Hungary
amjad.balawi20@gmail.com,abdullah.al.zoabi@outlook.com,{tomas.horvath,jlseixasjr}@inf.elte.hu

Abstract: Convolutional Neural Network (CNN) has become one of the most popular techniques in image classification. Usually CNN models are trained on a large amount of data, but in this paper, it is discussed CNN usage on data shortage and class imbalance issues. The study is conducted on a small dataset of vine leaves images on a classification task with five classes using two different approaches. In the first approach, a simple CNN model is used, while in the second approach, the Visual Geometry Group (VGG) model with transfer learning is used. It is shown that using different deep learning techniques such as transfer learning, stratified sampling, data augmentation, and the state of arts CNN models such as VGG gives a relatively very good model performance with up to 87% accuracy.

1 Introduction

Deep Learning (DL) was inspired by the human brain and try to simulate how humans learn. In DL, networks of neurons organized in multiple layers analyze large amounts of data to find the underlying structure or pattern, the main idea is to do that automatically without explicitly programming it, the computer learns how to classify text, sounds and images. In Computer Vision (CV) tasks, the computer is trained on huge amount of images by encoding these images pixels into internal representation, so the classifier can find the patterns on the input images [1].

DL outperforms other solutions in multiple domains, including speech, vision, video and natural language processing, it also reduces the use of feature engineering stage which is one of the most time-consuming tasks in machine learning [2]. The other reason, that made DL so famous in the last few years, is a huge improvement in terms of computational power that can be utilized to accomplish such tasks. However, one common problem is to preform badly on unseen data (test dataset), due to over-fitting, usually, a large dataset is required to increase the model performance. Another problem is that it is hard to choose the right model for any given problem.

Convolutional Neural Network (CNN or ConvNets) is a sort of Neural Network mostly popular in image classification [3] but it has a fewer number of connections, which means, a fewer number of model parameters making it less sensitive to over-fitting. The second reason why CNN is powerful in computer vision tasks is the parameter sharing, which means, if the filter is useful on a part of the image it could be useful on another one. Furthermore, CNNs preserves the spatial information of the image which makes the classifier more robust against the affine transformations like translation and rotation.

In many cases, especially in the current times, image data scarcity can be dealt by frequent acquisition, but there are still some situations in which acquisition is not easy or may not be frequent, as in agriculture, where a plant can not be created in an hour or a day. There are also cases where synthetic images creation is far from real world images, so training any model in this situation would create good controlled results but would not solve real problems.

The goal of this article is to find techniques, procedures or functions that can deal with the problems of using CNNs in small and imbalanced databases. For such, two different structures of CNN are implemented, with combination of different DL techniques and procedures such as data augmentation, transfer learning, stratified sampling and model picking based on validation accuracy, also showing the transition from a simple CNN model to a state of art model like VGG.

This paper is organized as follow: Section 2 presents the techniques and definitions used in the proposals of this work, followed by Section 3 which describes the steps for constructing the models. Section 4 shows the results obtained and in Section 5 the conclusions that can be inferred.

2 Proposed Approaches

There are many Machine Learning (ML) techniques that could be used for general classification problems like K-Nearest Neighbor (KNN), Logistic Regression, Support Vector Machines (SVM) and Artificial Neural Networks (ANN), but in term of the image classification problems the most popular technique is the Convolution Neural Networks. CNN is a class of ANNs that has become dominant

in various CV tasks [4], due to its ability to extract relevant features from raw data [5].

2.1 CNN and VGG architectures

In general, the CNN architecture is like an ordinary Neural Network, but it is stronger and deeper because it preserves the spatial information of images to overcome the problem of affine transformations. It also makes the classifier more robust by adding a stack of convolution layers just before the dense layers, besides it reduces the number of trained parameters which speeds up the learning process. CNN architecture includes several building blocks, such as convolution layers, pooling layers, and fully connected layers. A typical architecture consists of repetitions of a stack of several convolution layers and a pooling layer, followed by one or more fully connected layers [4].

Figure 1: Overview of the CNN architecture.

Figure 1 shows a general overview of the CNN architecture. Convolutions layers take the raw image as an input, perform convolutions using different sized trainable sliding windows which are typically named kernels and produce a vector which goes as an input for the dense layers. Each kernel has its own parameters which are trained just like the dense layer parameters, the output of convolutions layer goes as input to the next layer which looks for a higher level of input details and so on. The pooling layers come after a stack of one or more convolution layers, the purpose of pooling is to reduce the input size and overcome the small translations, there are multiple types of polling like Average, Min and Max polling.

The Visual Geometry Group (VGG) network was introduced by Simonyan and Zisserman [6] and is, in general, characterized by its simplicity since its only using 3×3 convolution layers on top of each other with increasing depth. In order to reduce the volume size or resolution, max-pooling was used in this network. After the convolution layers, there are two dense layers with 4,096 neurons each, followed by a softmax classifier, which is a generalization of the logistic regression to support the multiclass probability distribution. There are two version of VGG, 16 and 19, referring to the number of weight layers in the network.

Simonyan and Zisserman found the convergence of VGG16 and VGG19 on the deeper networks quite challenging so they trained smaller versions of the model as the one shown in Table 1. The main drawbacks with VGG network it is slow to train and weights are quite large. Due to the depth and the number of fully connected neurons makes it require a large amount of memory which makes it a tedious task. However, in this paper, we suggested methods to overcome this issue and speeding up the training process.

Table 1: VGG architecture

Convolution network configuration	
11 weights layer	16 weights layer
Input (224×224) RGB image	
Conv3-64	Conv3-64 Conv3-64
Max pooling	
Conv3-128	Conv3-128 Conv3-128
Max pooling	
Conv3-256 Conv3-256	Conv3-256 Conv3-256 Conv1-256
Max pooling	
Conv3-512 Conv3-512	Conv3-512 Conv3-512 Conv1-512
Max pooling	
Conv3-512 Conv3-512	Conv3-512 Conv3-512 Conv1-512
Max pooling	
FC-4096	
FC-4096	
FC-1000	
SoftMax layer	

2.2 Stratified Sampling

Stratified sampling is a probability sampling technique that takes the group size into account while doing the sampling process. The elements in target population are divided into distinct groups or so-called "strata", where within each stratum, the elements have similar characteristics to each other [7]. This technique is used widely in ML especially when the data suffers from class imbalance issue [8, 9, 10, 11]. This sampling technique is implemented in the scikit-learn library which is a free ML library for python. Sampling technique was used while splitting the data into train, validation and test sets using the attribute stratify inside *train_test_split* function and defining the target variable from which the sample was required.

2.3 Data Augmentation

DL models, including CNNs, are usually trained on a large amount of data to have a reasonable performance [12], in case of data shortage, like in this paper, these models tend to over-fit training data and lose the generalization ability

which leads to bad performance on the test dataset. After the cleaning stage, our dataset contains around 1600 images, training data was 80% of those images, while the reaming 20% were divided equally to testing and validation datasets. Roughly, this amount of data may not be enough to train a deep neural network and produce a good accuracy, thus to increase the accuracy, generalization and prevent over-fitting a data augmentation stage was added to the architecture.

Data augmentation means to create more training images based on the existing ones by applying some simple effects and affine transformations like shifting, flipping, rotating, zooming and so on. This augmentation will increase the number of training images and leads to more generalization and smoother training curve, it also provides information on small deformations images may contain due to acquisition processes [13]. Figure 2 shows the result of applying the data augmentation on a the first image resized to 256×256 which produced the second and third images by applying rotation and flipping.

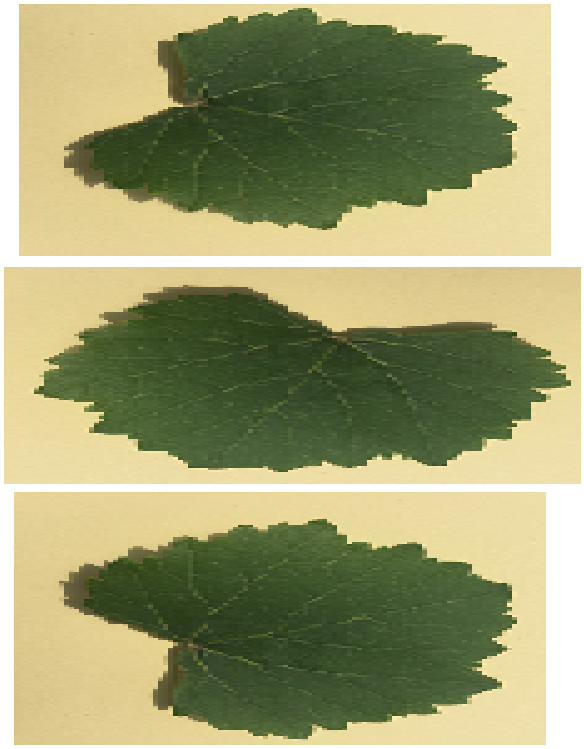

Figure 2: Example of Data Augmentation after Resizing the Original Image to 256×256.

As possible to see, some important shapes or features for classification that could be discarded if the acquisition was made only with the leaf upright, now also becomes part of training set.

2.4 Transfer Learning

Transfer Learning is widely used in machine learning when there is not enough data for model training and the main idea of this technique is to use a pretrained model which was trained on a similar problem, then apply this model on the new problem [14]. In most cases, the last few layers are refined and a simple dense or a linear model added on top of that.

ImageNet dataset was used in this paper, which is a large visual dataset designed for object recognition tasks which contains more than 14 million images and have been hand-annotated to indicate what objects are pictured in at least one million of the images, bounding boxes are also provided [15, 16]. ImageNet contains more than 20 thousand categories with typical categories, such as “balloon” or “strawberry”, consisting of several hundred images [17].

3 Research Methods

All strategies were implemented on Google Colab cloud service using Tensorflow 2.0 GPU and Keras API abstraction framework. Tensorflow is one of the famous libraries that is commonly used for image classification in DL. Tensorflow is an end-to-end open source software ML platform developed by the Google in 2015 for numerical processing and computation. Keras is an open source neural-network library written in python, with the main purpose of simplify code complexity, it also offers a simple/efficient API able to run on top of Tensorflow, Theano and other DL frameworks.

3.1 Dataset creation

In this study, images were collected by our department from the fields of Hungary in the summer of 2019. This study has an industrial background in the wine production and the purpose is to predict the type of wine produced by each vine. Around 2200 images were collected by different people and devices which produced images with different sizes, formats and background, so filtering and preparation stage was needed. The dataset is divided into five classes, each class is named in Hungarian after the wine produced from the tree as “Cabernet Franc”, “Kékfrankos”, “Sárgamuskotály”, “Szürkebarát”, and “Tramini”. Figure 3 shows eight random samples from dataset with their original sizes.

The two main problems faced and discussed in this study are data shortage and class imbalance, and both of them can be seen from histogram presented in Figure 4, which shows how many images there are in the dataset for each class.

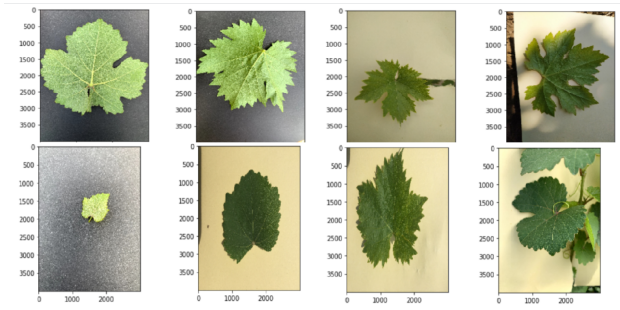

Figure 3: Random samples from the Dataset with their Original Sizes.

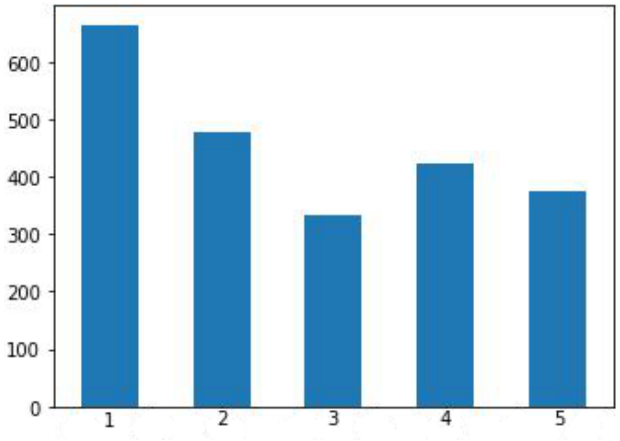

Figure 4: Histogram of the Raw Dataset.

Since data were collected by non experts and this is the first time using it, the first step was to clean this dataset by removing noisy images, as shown in Figure 5, so it would not affect the training process in a small dataset, while Figure 6 shows the distribution of the cleaned dataset.

Figure 5: Example of a Noisy Image.

Then all the different images format were unified into a common format (PNG), which was selected to keep as much information as possible in the images since its uses a lossless compression algorithm. After that, the images

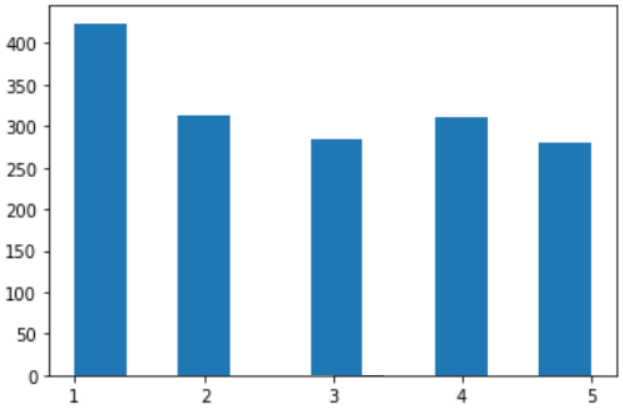

Figure 6: Histogram of the cleaned Dataset.

were resized into two resolutions 224×224 and 256×256 pixels which are practically preferred by different CNN architectures such as VGG16 and ResNet34. In order to speed up the training process, the raw images were converted into NumPy which is a vectorized implementation. Figure 7 shows an image sample from cleaned dataset.

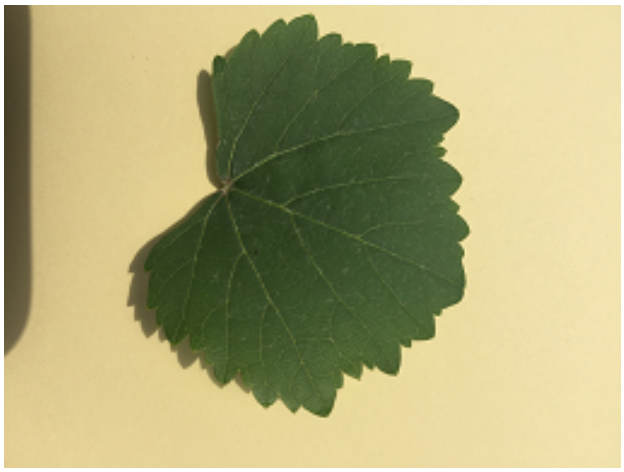

Figure 7: Example from the Prepared Dataset Resized to 256×256.

As it is noticeable from histogram, the dataset is relatively small, especially for deep learning models and the data suffer from the imbalance classes issue. So, in order to tackle these issues, data was split into training, validation and testing sets using stratified sampling, which takes samples from each class proportional to the class size [7].

The split used in the experiments was 80%-10%-10% for the training, validation (which is used for hyper-parameters tuning) and testing sets respectively. We used this split because the data is relatively small and we incorporate the stratified sampling which took the samples proportional to the class size for better generalization. After splitting, the data was normalized using MinMax scaler in order to speed up the training process by making the objective function more round, smooth and easy to optimize [18].

3.2 Simple CNN Model

This architecture was built by trial and error starting from a straightforward model inspired by LetNet-5 [19] architecture.

The first model consisted of two sets of one convolution and one pooling layers followed by two dense layers, but it showed bad accuracy due to under-fitting. So, layers were added, one layer per experiment, until no improvement was detected.

Then, multiple experiment were made by trying different combinations of kernel sizes, hidden layers sizes and pooling types. The best accuracy-wise model based on the two classes classification performance as the following:

- Three convolution blocks with 4, 8, and 16 filters.
- Each block consists of two convolutional layers followed by a Max pooling layer.
- Stack of three dense layers of 64, 32 and 5 units each.

3.3 VGG

Like the simple model, some attempts have been made for a better starting point. In the case of the VGG model, the Transfer Learning technique using the ImageNet dataset was the very first step and, from different experiments, it was noticeable that training only the last few layers of VGG model would provide the best results.

The reason for this behavior is that, in CNNs, the first few layers capture the low-level features which in most cases are useful in image classification issue. However, the last few layers are capturing the high-level features which are, in most cases, dataset (problem) specific. At the top of the model, the 1000 classes were removed which are related to ImageNet dataset and added the last dense 5-classes layer. Adam optimizer with 0.001 learning rate was also used.

The other technique used to handle the class imbalance issue was data augmentation on the training set. For reproducibility purposes, random seed was set while splitting the data into training, validation and test sets and the model weights with the lowest validation loss was saved using HD5 format.

4 Results

For the simple CNN model, the best result obtained among all experiments was 90%, 90% and 90% for Accuracy, Precision and Recall, respectively on the pair of classes "Szürkebarát" and "Tramini". This method of training was chosen to start as it is not time consuming and gives us the ability to do more trials. Also, this way enables the division of the five classes dataset into multiple two classes datasets and monitor the model performance among them.

Overfitting is noticeble from Figure 8, but at this point there was no need to seek improvement since two classes

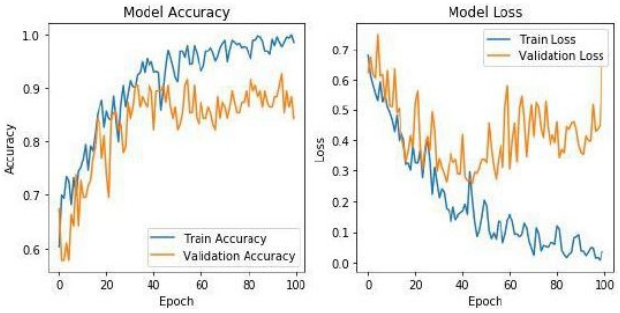

Figure 8: Model Performance in Two Classes.

classification was not the intended classification, a robust model was rather interesting. While verifying the model in four classes, two problems were faced, huge over-fitting and the largest class tend to have a large number of False Positives which leads to bad Precision and Recall. At this point, some steps were taken to smooth the effects of the problems:

- Increased the number of epochs to 300.
- Every 50 epochs, the train and validation datasets were merged and split randomly again to train and validation datasets.
- While training, the model was saved from the epoch with best validation accuracy. At the end, it was compared with the final model based on the test accuracy.

Among all the experiments with four classes, the best result were 88.4%, 88.4% and 88.1% for Accuracy, Precision and Recall, respectively. Figure 9 brings the performance of the model while training with four classes, based on training and validation accuracy and loss through the epochs.

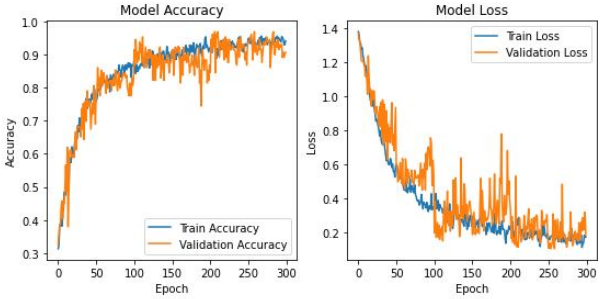

Figure 9: Model Performance in Four Classes.

Finally, the model was trained with five classes and the best results among all experiments where 83.8%, 84.4% and 84% for Accuracy, Precision and Recall.

Figure 10 shows the same information as Figure 9 while training the model with all five available classes using the simple model.

While for the VGG model, some transformations (width shift, height shift, zooming, shearing and rotation) were used in Data Augmentation, which led the model to

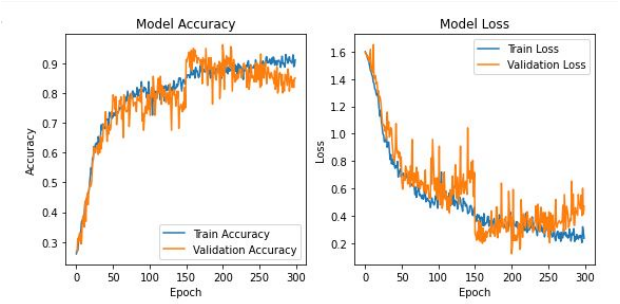

Figure 10: Model Performance in Five Classes.

achieve almost 87% accuracy on the test set, which served as an unbiased estimator. Precision, Recall and F1-score also reached about the same value.

Class	Precision	Recall	f1-score
0	0.89	0.93	0.91
1	0.82	0.90	0.86
2	0.92	0.79	0.85
3	0.93	0.84	0.88
4	0.80	0.86	0.83
accuracy			0.87
macro avg	0.87	0.86	0.87
weighted avg	0.87	0.87	0.87

Table 2: Precision, Recall and F1-score of the model

Table 2 shows the Precision, Recall, and F1-score using the VGG model. The metrics used to measure the model's performance were chosen considering they take into account the class imbalance issue and the general intuition behind them, that precision means how much noisy data is provided, in other words, it is more related to False Positive rates, while recall means how much good data is missed, and finally the f1-score is the harmonic mean of precision and recall. The main reason that harmonic mean used in f1-score is to punish the large difference between precision and recall. For example, if there were 100% precision and 0% recall, the f1-score will be 0%, while the arithmetic mean would be 50%.

5 Conclusion

In this research, we investigated different deep learning techniques to overcome data shortage and class imbalance issues. With experiments, we noticed that even the deep leaning models which require a lot of data can be performed very well even on a small imbalanced dataset using techniques such as stratify sampling, data augmentation, and transfer learning. In our first experiment, which is using a simple CNN model we got an accuracy around 83.8% and almost the same for other metrics (Precision, Recall, and F1-score), while in the second experiment a VGG model was used with a combination of different techniques reaching very good results of about 87% for the accuracy and other metrics.

Results indicate that even if a large amount of data is preferable, it is possible to overcome the previously mentioned issues with satisfactory results. In addition, the applied techniques contributed to non-appearance of overfitting, making the models not database dependent.

It is also possible to realize that, in cases where the required level of accuracy is very high, above 90% or 95%, the techniques applied may not be recommended without further database analysis, since these techniques may sacrifice accuracy to avoid other problems.

Also important to notice that one of the models is already known in literature and the other did not required any major framework to be built, only applying systematic and incremental analysis while interpreting obtained results during each step.

Acknowledgement

We would like to thank Telekom who has us as one of its technology partners on Telekom Innovation Laboratories and the Tempus Public Foundation for the financial support through the Stipendium Hungaricum Scholarship Programme.

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

References

[1] W. J. Zhang, G. Yang, Y. Lin, C. Ji, and M. M. Gupta. On definition of deep learning. In *2018 World Automation Congress (WAC)*, pages 1–5, 2018.

[2] Guillaume Chassagnon, Maria Vakalopolou, Nikos Paragios, and Marie-Pierre Revel. Deep learning: definition and perspectives for thoracic imaging. *European Radiology*, 30:2021 – 2030, 2019.

[3] Sakshi Indolia, Anil Kumar Goswami, S.P. Mishra, and Pooja Asopa. Conceptual understanding of convolutional neural network- a deep learning approach. *Procedia Computer Science*, 132:679 – 688, 2018. International Conference on Computational Intelligence and Data Science.

[4] Rikiya Yamashita, Mizuho Nishio, Richard Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9, 06 2018.

[5] J. Moreira, A. Carvalho, and T. Horvath. *A General Introduction to Data Analytics*. Wiley, 2018.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv arXiv:1409.1556v6 (ICLR 2015)*, 10 Apr 2015.

[7] Van L. Parsons. *Stratified Sampling*, pages 1–11. American Cancer Society, 2017.

[8] Elizabeth Tipton. Stratified sampling using cluster analysis: A sample selection strategy for improved generalizations from experiments. *Evaluation Review*, 37(2):109–139, 2013. PMID: 24647924.

[9] Kevin Lang, Edo Liberty, and Konstantin Shmakov. Stratified sampling meets machine learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2320–2329. JMLR.org, 2016.

[10] Longhua Qian, Guodong Zhou, Fang Kong, and Qiaoming Zhu. Semi-supervised learning for semantic relation classification using stratified sampling strategy. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1437–1445, Singapore, August 2009. Association for Computational Linguistics.

[11] Uchida S Goldstein M. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS ONE 11(4): e0152173*, 2016.

[12] Luke Taylor and Geoff Nitschke. Improving deep learning using generic data augmentation. *CoRR*, abs/1708.06020, 2017.

[13] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning, 2017.

[14] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, May 2016.

[15] *New computer vision challenge wants to teach robots to see in 3D*. New Scientist, 7 April 2017. Retrieved 3 February 2018.

[16] John Markoff. *For Web Images, Creating New Technology to Seek and Find*. The New York Times, Retrieved 3 February 2018.

[17] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[18] Ramírez-Gallego S. Luengo J. et all García, S. Big data preprocessing: methods and prospects. *Big Data Anal 1*, 1, 2016.

[19] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.