Advances in Science, Technology and Engineering Systems Journal Vol. 6, No. 1, 989-994 (2021)

www.astesj.com

ASTESJ

ISSN: 2415-6698

Classifying Garments from Fashion-MNIST Dataset Through CNNs

Alisson Steffens Henrique*,[1], Anita Maria da Rocha Fernandes[2], Rodrigo Lyra[2], Valderi Reis Quietinho Leithardt[3,4], Sérgio D. Correia[3,4], Paul Crocker[5], Rudimar Luis Scaranto Dazzi[2]

[1]*Master in Applied Computing, Univali, School of the Sea Science and Technology, Itajaí, 88302-901, Brazil*

[2]*Laboratory of Applied Intelligence, School of the Sea Science and Technology, Itajaí, 88302-901, Brazil*

[3]*VALORIZA, Research Center for Endogenous Resources Valorization, Instituto Politécnico de Portalegre, 7300-555 Portalegre, Portugal*

[4]*COPELABS, Lusófona University of Humanities and Technologies, Campo Grande 376, 1749-024 Lisboa, Portugal*

[5]*Departamento de Informática, Universidade da Beira Interior, Instituto de Telecomunicações, Delegação da Covilhã, 6201-601 Covilhã, Portugal*

ARTICLE INFO

Article history:
Received: 27 October, 2020
Accepted: 24 January, 2021
Online: 16 February, 2021

Keywords:
Image Classification
Deep Learning
Convolutional Neural Networks

ABSTRACT

Online fashion market is constantly growing, and an algorithm capable of identifying garments can help companies in the clothing sales sector to understand the profile of potential buyers and focus on sales targeting specific niches, as well as developing campaigns based on the taste of customers and improve user experience. Artificial Intelligence approaches able to understand and label humans' clothes are necessary, and can be used to improve sales, or better understanding users. Convolutional Neural Network models have been shown efficiency in image c1assification. This paper presents four different Convolutional Neural Networks models that used Fashion-MNIST dataset. Fashion-MNIST is a dataset made to help researchers finding models to classify this kind of product such as clothes, and the paper that describes it presents a comparison between the main classification methods to find the one that better label this kind of data. The main goal of this project is to provide future research with better comparisons between classification methods. This paper presents a Convolutional Neural Network approach for this problem and compare the classification results with the original ones. This method could enhance accuracy from 89.7% (the best result in the original paper, using SVM) to 99.1% (with a new cnn model called cnn-dropout-3).

1. Introduction

This paper is an extension of work originally presented in the Iberian Conference on Information Systems and Technologies [1].

The fashion market has changed dramatically over the last 30 years, resulting in an evolution in that industry [2]. Understanding customer tastes and better-directing sales are the way to increase profit [3].

The rise of internet business lets people buy their clothes through websites, faster and easier. The introduction of methods to improve user's experience when searching for items in these platforms is decisive [4].

Classifying clothes is part of the broad task of classifying scenes [5–9]. The automatic generation of image labels that describe those products can alleviate human annotators' workload [10]. This kind of information may also help labeling scenes and better understanding users' tastes, culture, and financial status [5].

In [11], the authors present Fashion – MNIST data set based on images from Zalando, which is the Europe's largest online fashion platform. Fashion MNIST has 70,000 products with 28x28 pixel grey scale images divided into 10 categories: t-shirt, trouser, pullover, dress, coat, sandals, shirt, sneaker, bags and ankle boots.

---

*Corresponding Author: Alisson Steffens Henrique, ash@edu.univali.br

https://dx.doi.org/10.25046/aj0601109

Those images were formerly thumbnails on their web store. This dataset is widely used in Artificial Intelligence (AI) benchmarks, and it is preloaded in Keras [12].

The original work used different AI models on this dataset to discover which one is better suited for the data labeling [11]. The evaluated implementations (from scikit-learn) were: Decision Tree, Extra Tree, Gaussian Naive Bayes, Gradient Boosting, k Neighbors, Linear Support Vector Classification, Logistic Regression, Multilayer Perceptron, Passive Aggressive Classifier, Perceptron, Random Forest, Stochastic Gradient Descent and Support Vector Classification. Their best result was using SVM, and they could achieve 89.7% of accuracy.

CNNs can have better results when compared to SVM [13]. Knowing that, this paper proposes the use of Convolutional Neural Networks (CNN) to label FashionMNIST dataset. The main goal is to compare those results with the original one, providing future research to be able to easily choose the most suitable classification method. In this paper, we present the development of four different CNN models and compared the results with the original ones. Our original work [1], used these four models with TensorFlow 1 (TF1) to get the results. Now, we present this extension using TensorFlow 2 (TF2) and GPU computing (with tensorflow-gpu and keras).

2. Background

2.1. Feature Learning

Machine Learning refers to computer systems capable of learning and modifying their behavior, in response to external stimuli or through experiences accumulated during their operation [14].

The main objective of Machine Learning is to generalize beyond the existing examples in training set, because regardless the amount of existing data, it is very unlikely that during the tests, the same examples will appear [15].

Conventional Machine Learning techniques are limited to processing natural data in its raw form. To build a model capable of doing pattern recognition, it is necessary to develop a feature extractor, which transform raw data into a representation that the classifier can detect [16, 17].

The group of methods that allow systems (based on Machine Learning) to discover the necessary representations for detecting and classifying raw data is known as Feature Learning [18].

These methods can not only learn how to map the feature to a result, but also to build the representation itself, often resulting in better performance than the representations developed by a specialist [19].

The growing scientific interest in this topic has been followed by a notable success, both in academia and industry. Mainly in areas of speech recognition, signal processing, object recognition and natural language processing [20].

Some of the Feature Learning techniques that have been producing promising results refer to Deep Learning. As more data becomes available, the more successful this technique will be [19].

2.2. Deep Learning

There are several techniques capable of Feature Learning, some are known as Deep Learning. They are models made by multiple nonlinear transformations, to produce abstract and more useful representations.

These models transform a representation into another, more abstract than the previous. In classification models, for example, more representation layers tend to amplify aspects of the data that are important for classification and hide irrelevant variations. By adding more layers, these models can represent more complexes functions [21].

The key aspect of Deep Learning is that layers of representations are not specified by a human specialist. They are learned through data, using common Machine Learning procedures. In the Deep Learning context, we can use Convolutional Neural Networks for this.

2.3. Convolutional Neural Network

Convolutional Neural Networks [22], refer to a variation of MLP (Multilayer Perceptron) and are based on the visual cortex behavior, where the neurons of the initial regions are responsible for detecting simple geometric shapes in the image, such as corners and edges, and the final neurons detect more complex graphic shapes. The process is repeated throughout the cortex until neurons in the final region detect characteristics of higher abstraction level, such as specific faces [23].

CNNs are used in problems where it is necessary to find relevant information implicit in data set, through operations that occur in convolution and pooling layers. In relation to the image classification task, variants of Convolutional Neural Networks that have been prevalent in the literature [24], and they demonstrate excellent results in the MNIST, CIFAR and ImageNet datasets [11, 25].

A convolutional layer is composed by several neurons, each one is responsible for applying a filter to a piece of the input matrix [23]. The convolution operation consists of applying a series of filters, sliding over the entire input matrix, and the result of applying the filters is called a feature map [22].

A pooling layer implements a nonlinear sub sampling function to dimensional reduction and small invariances capture. Pooling reduces the dimensionality of the input feature map and produces a new feature, creating something like a summary of the input.

For each filter, the highest value (max pooling) is selected, or the average (average pooling) is calculated. The pooling application speeds up training and reinforces CNN's strength in relation to position and size of most important characteristics of the training data.

2.4. Dropout

One of the most common challenges in training a Convolutional Neural Network is overfitting. There are several ways to mitigate this problem in a Deep Learning model: increasing the number or size of layers, or use a technique known as dropout.

The term dropout refers to "dropping" units (neurons) from a neural network, which means, temporarily removing them from model. The choice of which neuron to remove is random, and the amount can be fixed using a constant, for example, the constant 0.5 defines that half of the neurons will be removed.

This technique has considerably improved the accuracy and performance of neural network in several applications, such as object classification, speech recognition and document classification, among others [6]. This shows that this technique can be generalized for any problem.

3. Related Work

In [6], the authors presented a context sensitive grammar in an And-Or graph representation, that can produce a large set of composite graphical templates of cloth configurations.

In [5], the authors introduce a pipeline for recognizing and classifying clothes in natural scenes. To do so, they used a multi-class learner based on a Random Forest. Data was extracted features maps which were converted to histograms and used as inputs for a Random Forest (for clothing types) and an SVM (for clothing attributes) classifier. As result, their pipeline can describe the clothes on a scene, as show in Figure 1. They also created their own dataset with 80000 images labeled in 15 classes.

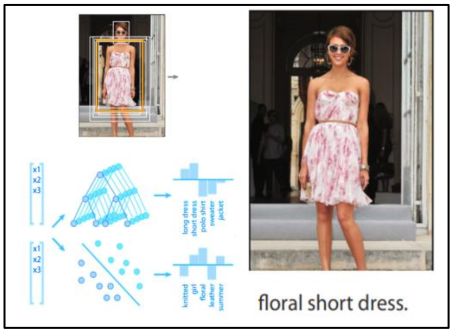

Figure 1: Model presented in [5].

In [26], the authors propose a knowledge-guided fashion analysis network for clothing landmark localization, and classification. To do so, they used a Bidirectional Convolutional Neural Network. As results, the model can not only predict landmarks, but also category and attributes, as shows Figure 2.

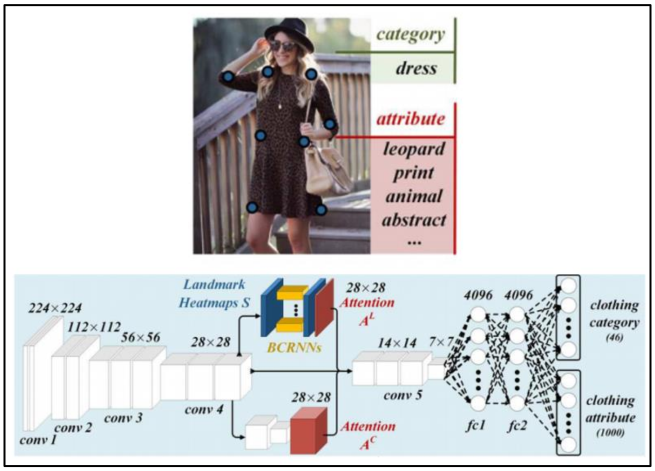

Figure 2: Model proposed [25].

4. Data Set

Fashion-MNIST is a direct drop-in alternative to the original MNIST dataset, for benchmarking machine learning algorithms [11]. MNIST [27] is a collection of handwritten digits, and contains 70000 greyscale 28x28 images, associated with 10 labels, where 60000 are part of the training set and 10000 of the testing. Fashion-MNIST has the exact same structure, but images are fashion products, not digits. A sample of this set can be seen in Figure 3.

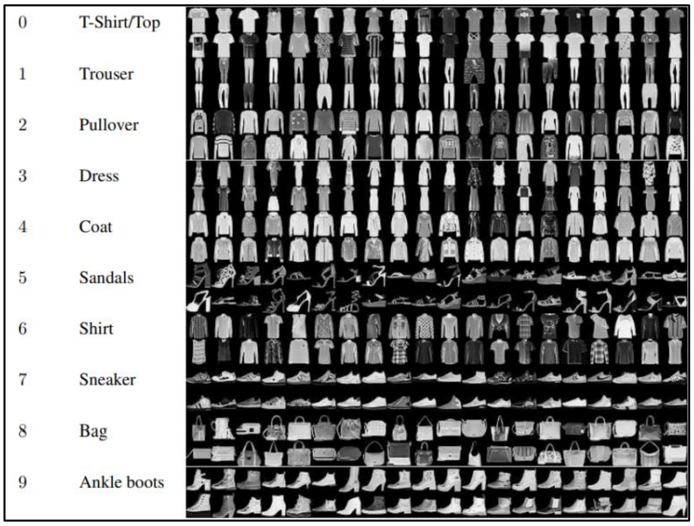

Figure 3: Fashion-MNIST sample

The dataset can be obtained as two 785 columns CSV, one with training images, and the other with testing ones. Each CSV row is an image that has a column with the label (enumerated from 0 to 9) and 784 remaining columns that describe the 28x28 pixel image with values from 0 to 255 representing pixel luminosity.

To make data access easier, we generated images divided in directories by usage, and labels. This way, we are able, to easily obtain image information by using only its path e.g., resources/test/dress/10.png.

5. CNN Models

To label this dataset, four CNN models were done in Python with Keras and TensorFlow. Training was executed in a Jupyter notebook, using GPU. We also used Weights and Biases [14] to grab information about training and hardware usage.

Proposed models were named: cnn-dropout-1, cnn-dropout-2, cnn-dropout-3 and cnn-simple. The goal of those models was to be able to label the dataset without the need of too much training or processing on activation, so developers can use it on real time applications such as online stores and searching websites.

5.1. cnn-dropout-1 and cnn-dropout-3

Both models use two consecutive blocks containing: a convolution, a max pooling, and finally a drop out. These blocks are connected to two more fully connected layers, who are connected to an output layer of ten neurons, each one representing a category. The only difference between these two models is that cnn-dropout-3 has considerably lower dropout values, as show in Figure 4.

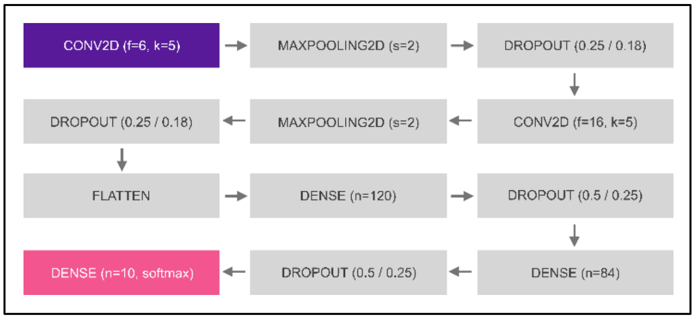

Figure 4: Topology of cnn-dropout-1 and cnn-dropout-3.

where the first drop out values is for model 1, and the second one for model 3. This topology has 44426 trainable parameters.

5.2. cnn-dropout-2

This proposed model is very similar to the cnn-dropout-1 model. However, it has two layers of convolutions before each max pooling. This model has about 32340 trainable parameters as shows Figure 5.

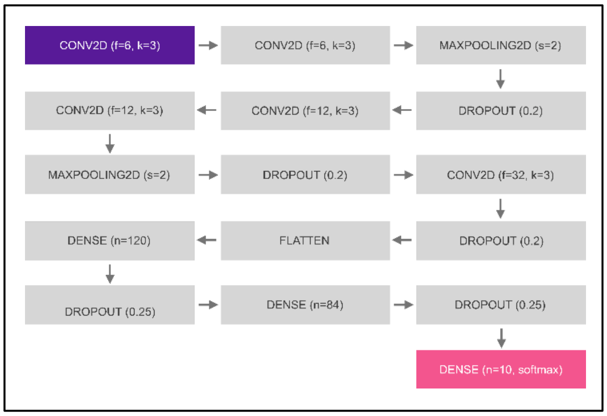

Figure 5: Topology of cnn-dropout-2.

5.3. cnn-simple

Cnn-simple is a model with less layers. It has only two convolutions, followed by a fully connected layer, in addition to the respective dropout and max pooling like other models. This model has 110968 trainable parameters and is shown in Figure 6.

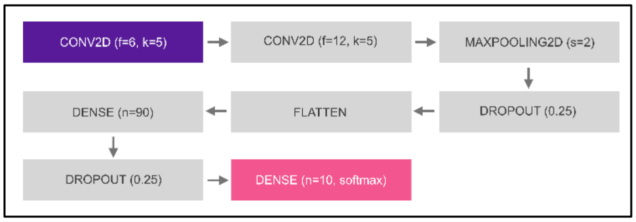

Figure 6: Topology of cnn-simple.

Since this model has only one max pooling, the image gets to the dense layer with 14x14 pixels in size (four times the size of other models which is 7x7). So, de dense layer training is expected to be slower.

All those models were modeled based on Keras Sequential model. Convolutional and dense layers used Rectified Linear Unit (ReLU) activation functions, except by the last dense layer on each model (output layer), were Softmax was used. The optimizer used was Adadelta [28] Batch size was 128 and we trained the models for 12 epochs. To improve results, image pixel luminosity values were normalized to float numbers between 0 and 1.

6. Results

On the training dataset, the most accurate model was cnn-simple, with 98.95% of accuracy. Figure 7 shows that other models were also acceptable, based on the results obtained: 98,06% (cnn-dropout-3), 97.51% (cnn-dropout-2), and even the worst model (cnn-dropout-1), got an accuracy of 96,46%.

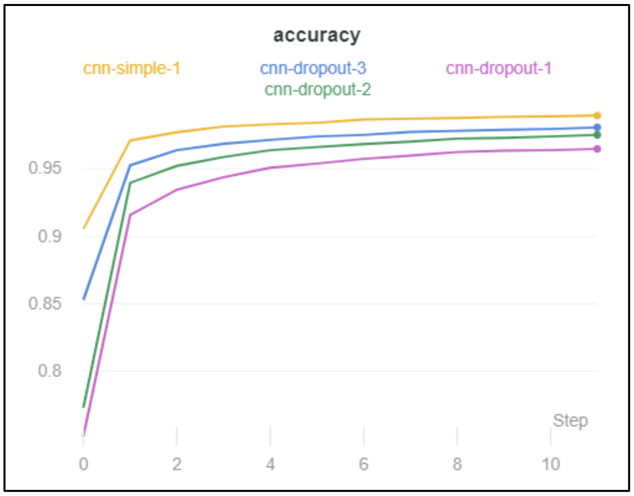

Figure 7: Accuracy

Validation accuracy had different results (as shows Figure 8), with cnn-dropout-3 having the best results 99.1%, followed nearly by cnn-dropout-2 (99.08%) and cnn-simple (99.05%). Cnn-dropout-1 still the one with the worst results (98.69%).

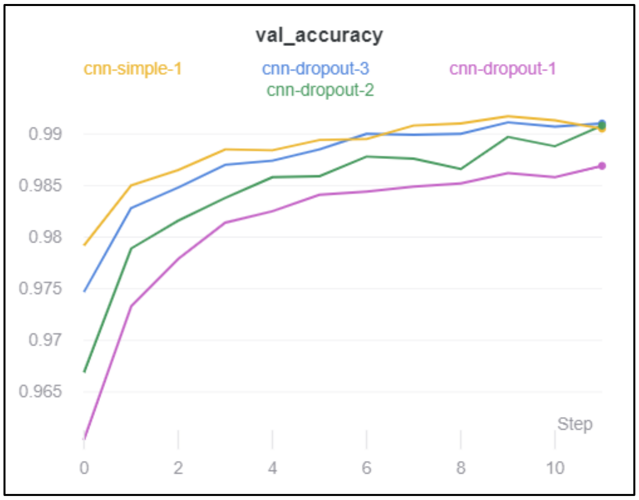

Figure 8: Validation Accuracy

Loss had good results, with cnn-simple having the best results both in training and validation loss, followed by cnn-dropout-3, cnn-dropout-2, and cnn-dropout-1. Results are shown in Figure 9.

When evaluating time, the two more accurate models, were also de faster in training. Figure 10 presents the relation between time and the training epoch. The lines tilt angle shows how fast the training was.

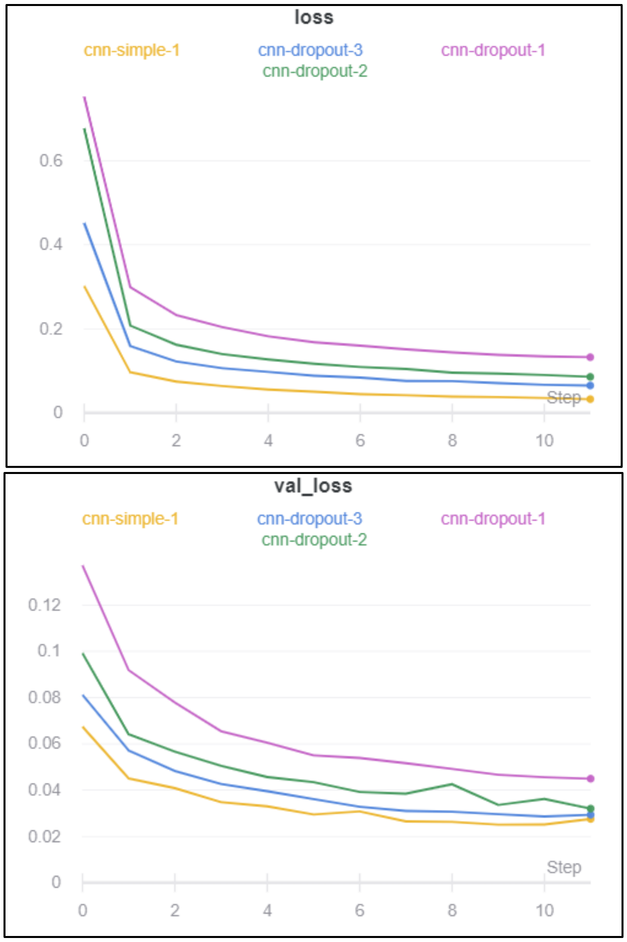

Figure 9: Training and Validation Loss

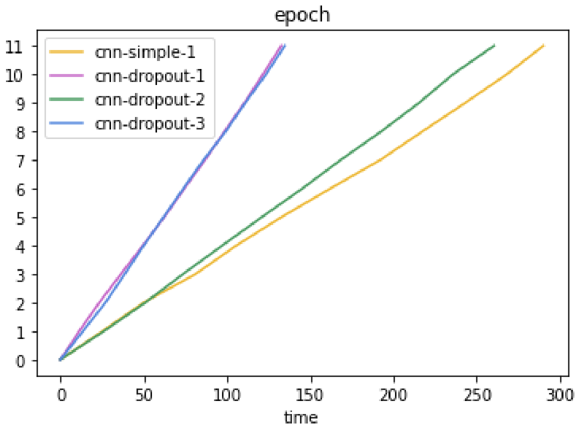

Figure 10: Training Time

Since we are using the same dataset, it is possible to compare these models with traditional non-convolutive machine learning algorithms presented in [11]. Table 1 presents the comparison among our model with theirs.

Table 1: Models Comparison

Model	Accuracy
cnn-dropout-3	99.10%
cnn-dropout-2	99.08%
cnn-simple-1	99.05%
cnn-dropout-1	98.69%
support vector classification	89.70%
gradient boosting	88.00%
random forest	87.30%
multilayer perceptron	87.00%
k neighbors	85.40%
logistic regression	84.20%
linear support vector classification	83.60%
stochastic gradient descent	81.90%
decision tree	79.80%
Perceptron	78.20%
passive aggressive classifier	77.60%
extra tree	77.50%
gaussian naive bayes	51.10%

Results shows that even our worst results (cnn-dropout-1) got better results than the best result in [11] (SVC, 89.70%).

7. Conclusions

Obtained results evidence that classifying fashion products with CNN can be more accurate than by using other conventional machine learning models. In addition, it was observed that the dropout technique together with more convolutive layers are effective when it comes to reducing the bias of a model.

Using TensorFlow 2 and GPU for training, we could reach not only a better training time, but also, better accuracies. Table 2 shows the differences between our original work and the present.

Table 2: Version Comparison

Model	**TF1**				**TF2**			
	Loss		**Accuracy**		**Loss**		**Accuracy**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
cnn-dropout-1	0.21	0.26	91.87	90.35	0.13	0.04	96.47	98.69
cnn-dropout-2	0.19	0.25	92.59	90.81	0.08	0.03	97.51	99.08
cnn-dropout-3	0.14	0.25	94.53	90.86	0.06	0.02	98.06	99.10
cnn-simple-1	0.04	0.26	98.91	91.72	0.03	0.02	98.95	99.05

We also could decrease loss and bias, which were our main problems. We could not evaluate improvements in runtime since we used different hardware than in the original run.

Our original work found that the best model was cnn-simple, but now, with these new results, we discovered that cnn-dropout-3 is better (using TF2). This is good news because this model is faster to train, since it has an extra max-pooling layer that decreases dense layer inputs by a quarter.

About our goals, we could compare obtained results with the ones from the original FashionMNIST paper, and they show that CNNs can be great classifiers for garments. Table 1 contains our main results about it and can be used in future works to help researchers and developers finding the best classification technique.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

This work was supported by the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) -

Financing Code 001, FCT/MCTES through national funds and when applicable co-funded EU funds under the project UIDB/50008/2020, and by Seed Funding ILIND - Instituto Lusófono de Investigação e Desenvolvimento, COPELABS [COFAC/ILIND/COPELABS 2020]. Al Proyeto: Uso de algoritmos y protocolos de comunicación en dispositivos con énfasis en la privacidad de los datos, and Laboratório de Telecomunicações de Portugal IT—Branch Universidade da Beira Interior, Covilhã.

References

[1] A. Hodecker, A.M.R. Fernandes, A. Steffens, P. Crocker, V.R.Q. Leithardt, “Clothing Classification Using Convolutional Neural Networks,” in Iberian Conference on Information Systems and Technologies, CISTI, IEEE Computer Society, 2020, doi:10.23919/CISTI49556.2020.9141035.

[2] R. Boardman, R. Parker-Strak, C.E. Henninger, “Fashion Buying and Merchandising,” Fashion Buying and Merchandising, 2020, doi:10.4324/9780429462207.

[3] Y. Zhong, S. Mitra, “The role of fashion retail buyers in China and the buyer decision-making process,” Journal of Fashion Marketing and Management, **24**(4), 631–649, 2020, doi:10.1108/JFMM-03-2018-0033.

[4] K.V. Madhavi, R. Tamilkodi, K.J. Sudha, “An Innovative Method for Retrieving Relevant Images by Getting the Top-ranked Images First Using Interactive Genetic Algorithm,” Procedia Computer Science, **79**, 254–261, 2016, doi:10.1016/j.procs.2016.03.033.

[5] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, L. Van Gool, “Apparel classification with style,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 321–335, 2013, doi:10.1007/978-3-642-37447-0_25.

[6] H. Chen, Z.J. Xu, Z.Q. Liu, S.C. Zhu, “Composite templates for cloth modeling and sketching,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, **1**, 943–950, 2006, doi:10.1109/CVPR.2006.81.

[7] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015, doi:10.1109/ICCV.2015.123.

[8] Z. Song, M. Wang, X.S. Hua, S. Yan, “Predicting occupation via human clothing and contexts,” Proceedings of the IEEE International Conference on Computer Vision, 1084–1091, 2011, doi:10.1109/ICCV.2011.6126355.

[9] K. Yamaguchi, M.H. Kiapour, L.E. Ortiz, T.L. Berg, “Parsing clothing in fashion photographs,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 3570–3577, 2012, doi:10.1109/CVPR.2012.6248101.

[10] K. Meshkini, J. Platos, H. Ghassemain, “An Analysis of Convolutional Neural Network for Fashion Images Classification (Fashion-MNIST),” in Advances in Intelligent Systems and Computing, Springer: 85–95, 2020, doi:10.1007/978-3-030-50097-9_10.

[11] M. Kayed, A. Anter and H. Mohamed, "Classification of Garments from Fashion MNIST Dataset Using CNN LeNet-5 Architecture," in 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE), Aswan, Egypt, 2020, 238-243, doi: 10.1109/ITCE48509.2020.9047776.

[12] A. Jain, A. Fandango, A. Kappor, TensorFlow Machine Learning Projects : Build 13 real-world projects with advanced numerical computations using the Python ecosystem, Packt Publishing Limited, Birmingham, United Kingdom, 2018. ISBN13: 9781789132212.

[13] Y. Shin, I. Balasingham, “Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification,” Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 3277–3280, 2017, doi:10.1109/EMBC.2017.8037556.

[14] S. Vieira, W.H. Lopez Pinaya, A. Mechelli, Introduction to machine learning, 2019, doi:10.1016/B978-0-12-815739-8.00001-8.

[15] O. Theobald, Machine Learning for Absolute Beginners, Scatter Plot Press, 169, 2017. ISBN: 1549617214.

[16] S. Shalev-Shwartz, S. Ben-David, Understanding machine learning: From theory to algorithms, Cambridge university press, 2013, doi:10.1017/CBO9781107298019.

[17] E.L. De Oliveira, “Machine learning techniques applied to predict the performance of contact centers operators,” Iberian Conference on Information Systems and Technologies, CISTI, 2019, doi:10.23919/CISTI.2019.8760665.

[18] J. Maindonald, "Pattern Recognition and Machine Learning," Journal of Statistical Software, **17**, 2007, doi:10.18637/jss.v017.b05.

[19] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press Cambridge, 2016. ISBN: 9780262035613.

[20] A. Peña, I. Bonet, D. Manzur, M. Góngora, F. Caraffini, “Validation of convolutional layers in deep learning models to identify patterns in multispectral images: Identification of palm units,” in Iberian Conference on Information Systems and Technologies, CISTI, IEEE Computer Society, 2019, doi:10.23919/CISTI.2019.8760741.

[21] N. Buduma, N. Locascio, Fundamentals of deep learning : Designing Next-Generation Machine Intelligence Algorithms, O'Reilly Media, Inc., 2017. ASIN: B0728KKXWB.

[22] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature, **521**(7553), 436–444, 2015, doi:10.1038/nature14539.

[23] K. Fu, D. Cheng, Y. Tu, L. Zhang, Credit card fraud detection using convolutional neural networks, Lecture Notes in Computer Science, **9949**, 483–490, 2016, doi:10.1007/978-3-319-46675-0_53.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, “Going deeper with convolutions,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1–9, 2015, doi:10.1109/CVPR.2015.7298594.

[25] A. Baldominos, Y. Saez, P. Isasi, "A Survey of Handwritten Character Recognition with MNIST and EMNIST," Applied Sciences, **9** (15), 3169, 2019, doi:10.3390/app9153169.

[26] W. Wang, Y. Xu, J. Shen, S.C. Zhu, “Attentive Fashion Grammar Network for Fashion Landmark Detection and Clothing Category Classification,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 4271–4280, 2018, doi:10.1109/CVPR.2018.00449.

[27] L. Deng, “The MNIST database of handwritten digit images for machine learning research,” IEEE Signal Processing Magazine, **29**(6), 141–142, 2012, doi:10.1109/MSP.2012.2211477.

[28] E. M. Dogo, O. J. Afolabi, N. I. Nwulu, B. Twala and C. O. Aigbavboa, "A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks," in Proceedings of 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), Belgaum, India, 2018, 92-99, doi: 10.1109/CTEMS.2018.8769211.