DOI: 10.2507/33rd.daaam.proceedings.xxx

CLASSIFICATION OF TWO-DIMENSIONAL MECHANICAL PARTS USING A CONVOLUTIONAL NEURAL NETWORK

Irina Topalova & Yuliana Staeva

This Publication has to be referred as: Topalova, I[rina]; Staeva, Y[uliana] (2022). Classification of two-dimensional mechanical parts using a convolutional neural network, Proceedings of the 33rd DAAAM International Symposium, pp. xxxx-xxxx, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-xx-x, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/33rd.daaam.proceedings.xxx

Abstract

Image search, object recognition and classification are emerging as key components in modern automated and autonomous production systems that integrate artificial intelligence. The accuracy in recognizing and classifying these parts, regardless of their geometric transformations, determines to a high degree the accuracy of their manipulation and positioning by the flexible assembly. Based on a comparison of some modern methods for classification of machine parts, the choice of a method with in-depth training of a convolutional neural network is justified. In the presented article a model for classification of machine parts is proposed, which is based on deep training of a convolutional neural network. A model was presented and tested, with a variety of training strategies for the purpose of increased efficiency. The proposed model was based on application of Batch normalization, Gaussian Noise, Weight regularization, Image normalization and Early stopping. High classification accuracy has been achieved for a large training and testing sample. The experiment was conducted for four classes of machine parts having different spatial position and orientation, as well different shapes of the objects belonging to the same class. The parts are grouped in an appropriate training, validation and testing sample. The stability and efficiency of the model under variations of the hyper-parameters of the model have been proven, supported by experimental results.

Keywords: Mechanical parts; Classification; Convolutional neural networks; Deep learning.

1. Introduction

The application of computer vision in the automated production processes in the field of additive manufacturing, predictive maintenance, packaging inspection, quality control in product and components assembly, is emerging as a necessary factor in achieving innovative solutions.
Image search, object recognition and classification are emerging as key components in modern automated and autonomous production systems that integrate artificial intelligence, aiming to overcome the challenges of implementing innovations in the modern European economy [1]. Implementing machine learning and artificial intelligence is a key stone of contemporary innovations in production automation as discussed in [2]. One of the main factors determining the efficiency of operations in the automated assembly of products is the high-precision recognition of mechanical parts,

located in a spatial disorder on a production working area. The accuracy in recognizing and classifying these parts, regardless of their geometric variations, determines to a high degree the accuracy of their manipulation and positioning by the flexible assembly [3]. In order to achieve high accuracy of recognition and classification of such objects, as well as in order to create an adaptive and easily reprogrammable environment, the methods of deep machine learning are applied.

In this study, we apply deep learning of a Convolutional Neural Network (CNN), using a chosen training strategy, in order to achieve high efficiency, applying the simplest model. The study was conducted for a large training, validation, and test sample of various mechanical parts. The investigated mechanical parts belonging to the same class, differ in shape, size, profile and spatial orientation. Different cases were studied, representing different hyper-parameter setups for the CNN. The analysis of the results and recommendations for future development and implementation of the model have been made.

2. State of the Art

There are various methods for classifying mechanical parts, integrated in real-time working hardware systems. Most of them use the "template matching" method, which “searches” for matching parts in the image with a predefined pattern/template [4]. The main disadvantage of this method is the need to pre-define the patterns and the delay due to the calculation of the correlation between the template and the corresponding area of the scanned image. The methods using neural networks are preferred, due to their adaptive abilities, the possibilities for deep training, and retraining in case of change in the setting of its parameters or in case of need for training with new mechanical parts. The question is what type of neural network is appropriate to use and optimise in the classification of mechanical parts? Submitting the entire image to the input neurons of a Multi-Layer Perceptron (MLP) network requires the design of a neural network with a large number of computational resources. It is necessary to pre-process the image in order to reduce its size, which in turn in many cases leads to loss of information. In this case, it is more appropriate to derive certain geometric or spectral features of the image, in order to constitute a feature vector to be used for training the MLP network [5]. It would be more appropriate to apply CNN, because the image size reduction is a basic idea that is implemented in the functional algorithm itself. Also, when training CNN, there is no need to group in advance different instances of mechanical parts, that belong to the same class (for example, class “bolt”, class “nut”, etc.) but differ in shape, size, profile, orientation, etc.

There are innovative solutions for the implementation of CNN neural networks for the classification of mechanical parts. In the implementation given in [6] in the pre-processing step, a quantum layer is added before the main CNN, which processes the data using quantum particle concepts. But this solution requires additional computing resources and proper data preparation before being submitted to a conventional CNN. The authors of [7] say CNNs are challenging to learn efficiently if the given dimension or amount of data or model become too large. We show in our research that training with a huge number of data/objects samples could give very good accuracy results together with a stable model work. The authors of [8] claim that CNNs trained on different popular datasets fail to detect objects when they see them under different lighting conditions and from different view angles. In this study, we show how by optimizing the parameters of the CNN, a very good result can be achieved even for such objects.

In our study, we aim to achieve maximum accuracy of recognition and classification of different classes of mechanical parts, without pre-processing of images and with conventional CNN. Various hyper-parameter setups have been applied, aiming to achieve the most simplified model while obtaining maximum efficiency. The research refers to training and testing with mechanical parts having different size, form, spatial orientation/different view angles, for exemplars belonging to the same class. The model is developed with coding in Python, using a cloud based platform.

3. General description, limitations and author’s approach

With this research we aim to solve a classification problem in the field of computer vision and object recognition systems, applied to machine parts with different 3D spatial orientation. Our proposed methodology includes the usage of Deep Learning Convolutional Neural Networks, aiming to reduce the necessity of image preprocessing. We introduce a variety of training strategies for the purpose of increased efficiency during the training process in terms of higher classification accuracy on the testing set of images, but achieving the most simplified model. The large training, validation, and test sample of various mechanical parts, belonging to the same class, but different in shape, size, profile and spatial orientation, is a good prerequisite for the generalization of the method and the model. The following research limitations were applied: no variant of CNN with a maximum or minimum number of convolutional filters was investigated, the optimal number was determined experimentally; objects with changed contrast, brightness or different illumination of the working area are not included in the training and testing samples, considering that in the general case these parameters are kept constant.

3.1 Studied objects

In order to implement a solution to the mechanical parts recognition problem, we are using an image dataset, which consists of 7404 images of mechanical parts. The dataset consists of four image classes, each representing samples of mechanical parts. The classes are: Bolt, Locating pin, Nut, and Washer. There is an equal amount of samples for each image class in the dataset. All samples were divided into training, validation, and testing sets, following the ratio:

Training samples - 70 % of total samples; Validation samples - 20 % of total samples; Testing samples - 10 % of total samples. Some sample images are provided in Table 1.

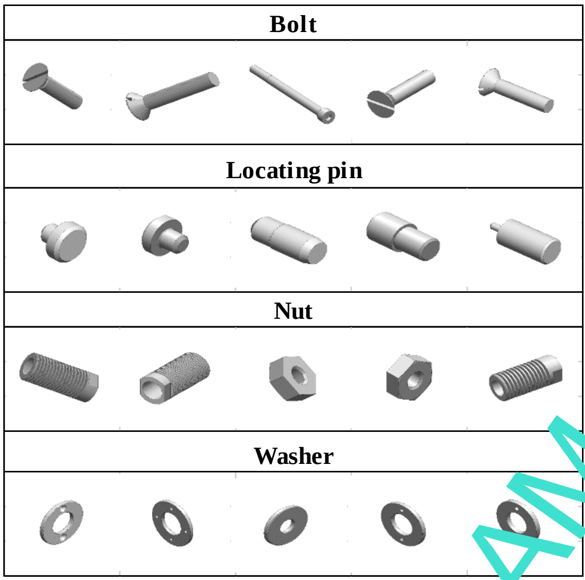

Table 1. Samples of the studied mechanical parts

3.2 The proposed Deep learning CNN model

The algorithm of choice for the solution of the image classification problem is a Deep Learning CNN. There are a total of four cases studied in this experiment, whereby each case represents a different hyper-parameter setup for the CNN. As the image samples are of size 224 x 224 pixels, we have decided to include only one convolutional layer, followed by one pooling layer before proceeding to flatten the convolved images. All four experimental cases are based on a simple topology due to the small image sizes. An increased number of convolution and pooling layers would result in greater dimensionality reduction of the images and loss of feature information. An overview of the primary network topology is shown in Fig. 1.

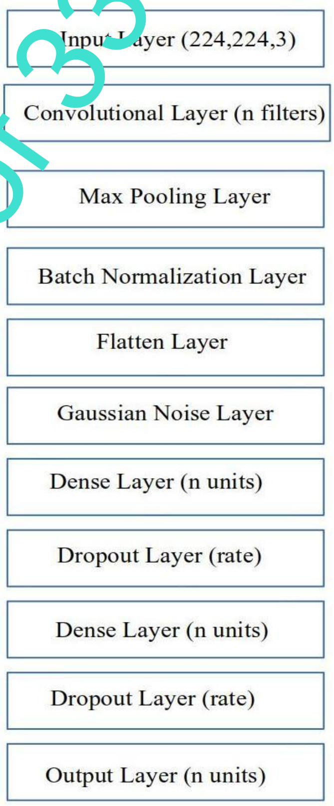

Fig. 1. Primary CNN topology

In order to increase the classification accuracy of all four CNNs, we have used the following training strategies:
Batch normalization: Consists of normalizing the input to each layer of the network, such that the inputs conform to a Gaussian distribution with a mean of zero and a standard deviation of one. This results in accelerated training for deep CNNs [9]. In this experiment the constructed CNN is not complicated in terms of depth, however, the usage of the batch normalization layer significantly decreases the number of training epochs required.
Gaussian Noise: this training technique is used to add random Gaussian noise to the image features with the aim to promote robust feature extraction. Gaussian noise is normally distributed and it has a mean of zero and a standard deviation of one [9].
Dropout: this training strategy resembles the bootstrap-aggregation training technique, whereby a number of simpler classifiers is used to solve a complicated problem, by dividing it into several less complicated problems. In the context of Neural Networks, a dropout layer is placed between two fully connected dense layers and it forces them to purposely ignore a random percentage of the input weights. This percentage is referred to as the dropout rate. The usage of a dropout rate results in preventing the neural network dense layers from trying to correct mistakes in previous layers, which in turn prevents the model from overfitting to the training data [9].
Weight regularisation: this training technique penalizes large weights by calculating either the sum of the values of the absolute weights (L1 norm) or calculating the sum of the squared weight values (L2 norm). The objective is to reduce the complexity of the produced decision boundary of the neural network and to keep the model as simple as possible to prevent overfitting. For this experiment, we have decided to use the L2 norm for weight regularisation [9].
Image normalization: this training technique reduces the computational complexity of training a Deep Learning CNN classifier, by rescaling the input data. In this case, the input data is in the form of images with pixel values in the interval [0, 255]. During the rescaling process, each pixel value is divided by 255. The output image has standardized values between [0,1] [9].
Early stopping: this training strategy includes monitoring the CNNs performance based on a chosen performance metric on each training iteration and interrupting the training process in case of successive poor performance [9]. For this experiment, the performance metric of choice is the validation accuracy.
The algorithm is implemented in Python in a cloud environment. The main parameters of the model are coded as follows.

```
cnn = Sequential([
            InputLayer(input_shape=(224,224,3)),
            Conv2D(filters =15 , kernel_size = (3,3), padding = 'valid', activation = 'relu', kernel_regularizer = l2(0.0005),strides=(2,2)),
            MaxPool2D(pool_size=(3,3)),
            BatchNormalization(),
            Flatten(),
            GaussianNoise(0.001),
            Dense(units = 20, activation='tanh', kernel_regularizer = l2(0.0005)),
            Dropout(rate = 0.1),
            Dense(units = 10, activation = 'relu', kernel_regularizer = l2(0.0005)),
            Dropout(rate = 0.1),
            Dense(units = 4, activation = 'softmax')
early_stop = EarlyStopping(monitor='val_accuracy',verbose = 1, patience = 30)
model_checkpoint = ModelCheckpoint('best_mech_cnn_2.h5', monitor = 'val_accuracy', verbose = 1, save_best_only = True).
```

3.3 Experimental setup

The experimental setup described in this article consists of the construction of four CNNs, each with a different hyper-parameter setup. The hyper-tuning of each CNN is given in Table 2:

Hyper-parameter setup	**Case 1**	**Case 2**	**Case 3**	**Case 4**
Convolutional filters	15	20	15	15
Kernel regularizer	0.0005	0.0005	0.0005	0.0005
Gaussian noise	0.001	0.001	0.001	0.001
Dropout rate	0.2	0.2	0.2	0.5
Learning rate	0.00007	0.00007	0.00007	0.00007
Strides	(2,2)	(2,2)	(1,1)	(1,1)
Total parameters	411454	548514	1643554	1643554

Table 2. Hyper-parameter setup for all experimental cases

4. Experimental results

In this section we present the experimental results. The results contain a graphical representation of the learning process of the CNN, with respect to accuracy and loss functions, as well as a table summary of highest to lowest classification accuracy.

4.1 Hyper-parameter influence

This section is focused on evaluating the graphical results from the neural network training process. The results from each case study are shown in Fig 2.

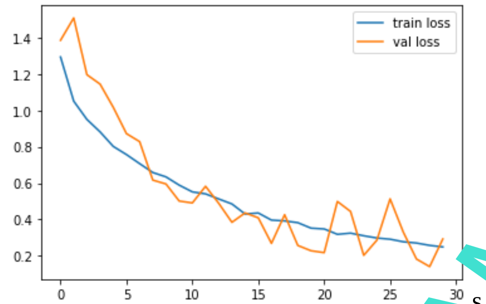

Fig. 2/a Loss function in case 1

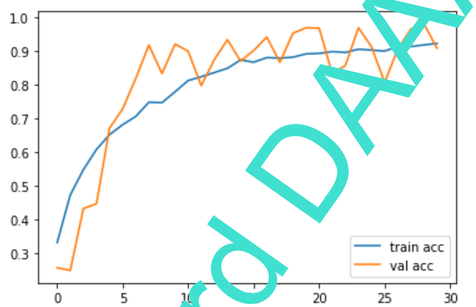

Fig. 2/b Accuracy in case 1

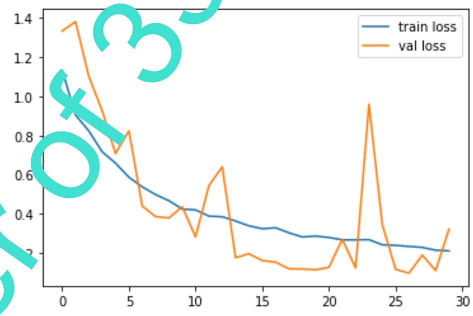

Fig. 2/c Loss function in case 2

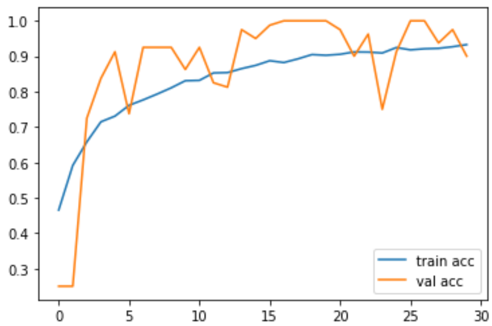

Fig. 2/d Accuracy in case 2

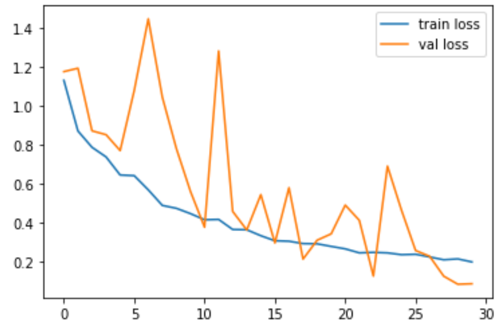

Fig. 2/e Loss function in case 3

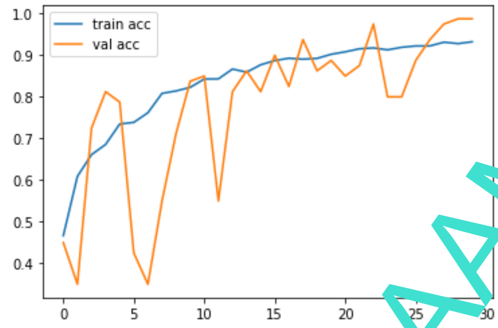

Fig. 2/f Accuracy in case 3

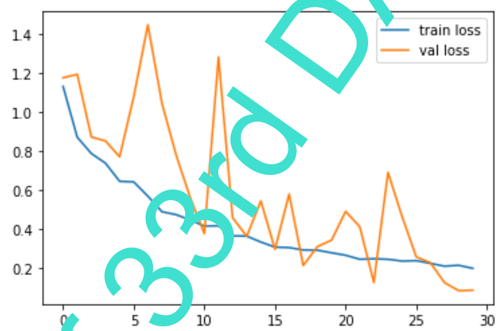

Fig. 2/g Loss function in case 4

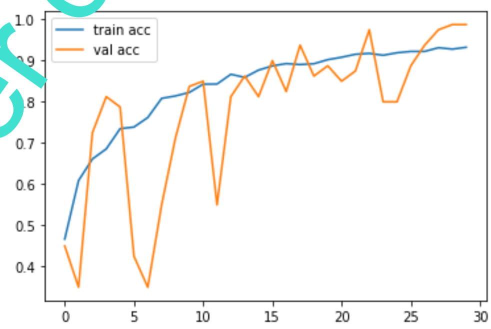

Fig. 2/h Accuracy in case 4

The first observation we have on the experimental results is that the stability in the training process decreases as the number of model parameters grows. In case 1 and case 2, both loss and accuracy functions of the validation set are similarly distributed to the loss and accuracy functions of the training set. This tendency decreases in cases 3 and 4, whereby the number of model parameters is increased due to the reduced strides of the convolutional filters. As a result, the validation loss and accuracy spike throughout the training process, making it unstable and less certain of a good outcome for both training and validation data.

Further observation in cases 1 and 2, shows that the increased number of filters in the convolutional layer has led to the model overfitting on the validation data in case 2. We assume that this is due to the fact, that the given image data is less complicated, with a clear contrast between the object to be recognized and the image background. Objects are visible, have clear edges and multiple angles of each object are included in the dataset. The simplicity of the images allows for effective feature extraction with fewer filters in the convolutional layer. Furthermore, this results in simpler network topology and therefore, fewer model parameters. The model has a reduced chance of overfitting the training data in case 1.

Performance comparison of the models in cases 3 and 4 reveals, that by increasing the dropout rate in case 4, the number of training epochs required to learn the training dataset is increased. Both models have the same amount of model parameters and a number of training epochs, however, in case 4 the accuracy on the training data only did reach 93 %, meaning that the model has not learned completely the training data.

4.2 Accuracy evaluation

In this section, we have provided a summary of validation accuracy for all four experimental cases. The result summary is shown in Table 3.

Case number	Accuracy
Case1	98.228 %
Case2	100 % (Overfitting)
Case3	98.750 %
Case4	93.750 %

Table 3. Result summary on validation accuracy for all 4 experimental cases

In order to study the model performance further, we have created a confusion matrix on each experimental case using the test dataset. The classification accuracy on each case is measured with the formula (1):

$$\text{Accuracy} = \mathit{Number\ of\ correctly\ classified\ samples}\,/\,\mathit{Number\ of\ classified\ samples} \qquad (1)$$

The classification accuracy on the test dataset in all four cases is provided in Table 4.

Case 1	**Case 2**
Accuracy: 97.31 %	Accuracy: 98.29 %
[[355 25 1 0]	[[370 11 0 0]
[1 376 2 2]	[7 370 1 3]
[0 7 374 0]	[0 2 379 0]
[2 1 0 378]]	[0 0 2 379]]
Case 3	**Case 4**
Accuracy: 99.47 %	Accuracy: 97.05 %
[[380 1 0 0]	[[381 0 0 0]
[4 374 2 1]	[15 355 4 7]
[0 0 381 0]	[6 1 365 9]
[0 0 0 381]]	[3 0 0 378]]

Table 4. Confusion matrix and classification accuracy on test data in four experimental cases

5. Conclusion

The analysis of the results, obtained when testing the proposed model, shows that the application of Batch normalization, Gaussian Noise, Weight regularization, Image normalization and Early stopping, allows simultaneously significantly reducing the number of training epochs required, promoting robust feature extraction and preventing the model from overfitting to the training data, at the same time producing a model as simple as possible to prevent overfitting. The achieved stable high accuracy of classification for the four cases of given hyper-parameters, shows that the proposed CNN model achieves the highest accuracy in case 3, although at the expense of increased number of parameters due to the reduced "stride" value. The achieved stability of the model and the high accuracy achieved for the variation of the parameters in the four presented cases show its efficiency in recognizing machine parts and can be recommended for classification of other two-dimensional objects. It was proven, that training with a huge number of data/objects samples could give very good accuracy results together with a stable model work, when appropriate CNN's hyper-parameters are adjusted. The represented experimental results show, that by optimizing the parameters of the CNN, a very good result can be achieved even for images of mechanical parts, having low resolution and different form, size, view angle and spatial orientation within the same class. As a future development of the method, we envisage conducting a larger number of tests with additional different classes of mechanical parts, as this would show further generalization of the model. Also, other possibilities for optimizing the parameters of the CNN may be sought. The same object samples could be tested for recognition and classification with adding a quantum layer before the main CNN and processing the data using quantum particle concepts, for benchmarking purposes.

6. References

[1] Pavlov, P. (2015). Innovation management, Book, University Publishing House of VFU (Varna Free University) "Chernorizets Hrabar", Varna, ISBN 978-954-715-645-6; Павлов, П. (2015). Иновационен мениджмънт, Учебник, Университетско издателство на ВСУ „Черноризец Храбър, Варна 2015, ISBN 978-954-715-645-6.

[2] Tekic, Z; Cosic, I & Katalinic, B. (2019). Manufacturing and the Rise of Artificial Intelligence: Innovation Challenges, Proceedings of the 30th DAAAM International Symposium, pp.0192-0196, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-22-8, ISSN 1726-9679, Vienna, Austria, DOI: 10.2507/30th.daaam.proceedings.025

[3] Nerakae, Ph., Uangpairoj, P. & Chamniprasart, K. (2016) Using Machine Vision for Flexible Automatic Assembly System, Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2016, pp. 428 – 435, 5-7 September 2016, York, United Kingdom, https://doi.org/10.1016/j.procs.2016.08.090.

[4] Topalova, I. (2016). Automated Visual Inspection System in Proline Industry, 27th International DAAAM SYMPOSIUM "Intelligent Manufacturing & Automation: Theory, Practice & Education", pp. 175-179, ISBN: 978-3-901509-70-4, ISSN: 17269679, Mostar, Bosna, October.

[5] Topalova, I. & Tatarska, V. (2018). Automated Classification of Similar Two-dimensional Objects with Neural Network Application, Proceedings of the 29th DAAAM International Symposium, B. Katalinic (Ed.), Published by DAAAM International, pp. 0225-0231, ISBN 978-3-902734, ISSN 1726-9679, Vienna, Austria, 24th-27th October, DOI: 10.2507/29th.daaam.proceedings.

[6] Khullar V., Tiwari R.G. & Agarwal A.K. (2022). Quantum Layer-Inspired Deep Learning for Mechanical Parts Classification. In: Agarwal B., Rahman A., Patnaik S., Poonia R.C., Proceedings of International Conference on Intelligent Cyber-Physical Systems. Algorithms for Intelligent Systems. Springer, Singapore, ISSN 978-981-16-7135-7, ISBN 978-981-16-7136-4.

[7] Seunghyeok Oh, Jaeho Choi & Joongheon Kim A. (2020). Tutorial on Quantum Convolutional Neural Networks (QCNN), 2020 International Conference on Information and Communication Technology Convergence (ICTC), 21-23 October 2020, Jeju, Korea (South), INSPEC Accession Number: 20244515, DOI: 10.1109/ICTC49870.2020.9289439, Publisher: IEEE.

[8] Manav Mandal. (2021). Introduction to Convolutional Neural Networks (CNN), Advanced Computer Vision Deep Learning Image Analysis, Published On May 1, 2021 and Last modified on july 23rd, 2021 https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/#h2_6.

[9] Goodfellow I., Bengio J. & Courville A. (2016). Deep Learning, Published by MIT Press, ISBN: 0262035618.