Military Vehicle Recognition with Different Image Machine Learning Techniques

Chapter · October 2020

DOI: 10.1007/978-3-030-59506-7_19

2 authors:

Daniel Legendre
MPKK
6 PUBLICATIONS **61** CITATIONS
SEE PROFILE

Jouko Vankka
The Finnish Defence Forces
114 PUBLICATIONS **1,345** CITATIONS
SEE PROFILE

Military Vehicle Recognition with Different Image Machine Learning Techniques

Daniel Legendre(✉) and Jouko Vankka

National Defense University, 00860 Helsinki, Finland
dnl.legendre@gmail.com, jouko.vankka@mil.fi
https://maanpuolustuskorkeakoulu.fi/sotatekniikan-laitos

Abstract. Different neural network training systems are studied for image recognition of military vehicles, variable start layer transfer training models and own convolutional neural networks training from scratch. Since, there is limited openly available military recordings, labeled social media images are used for training. Furthermore, expanding the image-set by random data transformation. An implementation is made in terms of image augmentation handling as an internal loop that freezes all numerical parameters of the neural network training, while selecting continuously a slightly larger section of the training set including an increment part of artificial images added to the system. All models where trained for three vehicle and two situational environment classification cases. The transfer learning is based on two of the most widely used recognition networks, ResNet50 and Xception, with a variable number of last trained layers to max. twenty. The first being successfully transfer-trained with validation accuracy values of ≈88%. In contrast Xception resulted on a over-fitted neural network with low validation accuracy and large loss values. Neither of the transferred schemes benefit from image augmentation. Moreover, in variable architecture training of convolutional networks, it was corroborated that different configurations of layers numbers/type/neurons adapt differently. Thus, a tailor-fit neural network combined with data augmentation strategy is the best approach with validation accuracy of ≈86.4%, comparable to large transferred networks with a ≈40 times smaller network architecture. Hence, requiring less computational resources. Data augmentation influenced an increment of validation accuracy values of ≈9.2%, with the least accurate network trained gaining up to 20% on accuracy due inclusion of artificial images.

Keywords: Machine learning · Neural networks · Optimization · Transfer learning · Image augmentation · 3D military machinery

1 Introduction

Several neural networks architectures exist nowadays for image recognition and classification, being one of the most useful for still single image classification,

A. Lopata et al. (Eds.): ICIST 2020, CCIS 1283, pp. 220–242, 2020.
https://doi.org/10.1007/978-3-030-59506-7_19

convolutional neural networks [24]. This has been extensible studied in civilian applications for animal classifications (cats vs dogs from Microsoft [18]), vehicles, face feature recognition, or general object classification [16]. One reason for the success for these recognition tasks are the large data sets available to train, test and validated different models [13,17]. To even further arrange different image recognition competitions with know data sets around the globe [14].

Nevertheless, even if this technology is available for military institutions, different challenges arise due to security and secrecy reasons from different military organizations, as it might not be convenient to share or even to record large databases (images) of diverse military equipment. Thus, arising different challenges in military applications, more centered on how to build a reliable training data base, with sufficient variety, to be used as training/validation set for deep convolutional or recurrent neural networks.

The rest of the present paper is structured as follows: Sect. 2 discusses related published works on the field of recognition of military equipment and applications, Sect. 3 presents the available data for machine learning training and their recollection sources, Sect. 4 explains the methods used in this research for expanding the available training data-sets, Sect. 5 shows the different hyper parameters used for the neural networks training schemes and clarify how it is ensured reproducible results, Sect. 6 explains the transfer learning schemes used and shows their performance to adapted targets, Sect. 7 explain the variable architecture scheme for trained from scratch networks, Sect. 8 presents a proposal on a new method to generate artificial images for different types of vehicles using real life locations and 3D designs, Sect. 9 display the performance of the different trained networks on testing images, to finally in Sect. 10 present an analysis of the different methods used in this investigation to train neural networks for military applications.

2 Related Work

There has been studies to adapt pre-trained models to military applications with limited data sets, such as Hiippala (2017) [11]. The study is mainly centered on variation of hyper-parameters optimal value using random search, yielding high accuracy values of 95% and above average accuracy ≈60% for convolutional network training, using cross-validation in their process, for both designs networks of image recognition. Those results show that training a convolutional neural network from scratch with augmentation is largely outperform by transfer learning architectures ≥35% [11]. Moreover, this paper attempts to take that study a level further and concentrate the training technique not on hyper-parameter optimization, but rather on neural network architecture and how to optimally select levels of training for a transfer learning algorithm and different architecture search for own design neural networks from scratch. Finally, to couple these algorithms with a traditional image augmentation technique [5,11] using Keras [2], and measure the level of influence that artificial created images have on different machine learning configurations. In order to prove that own design

architectures can be optimized to obtain high accuracy levels comparable to well transferred models by means of smaller computational requirements. Guo et al. (2016) surveyed the state-of-the art in deep learning algorithms in computer vision, and then briefly describes their applications in diverse vision tasks, such as image classification, object detection, image retrieval, semantic segmentation and human pose estimation [8]. The Convolutional Neural Networks (CNN) is the most extensively utilized and most suitable for images [8].

3 Military Vehicle Image Data Set

As an option to obtain training data from free accessible sources, social media sources are used. For the present task, these images where collected from previous works [11] and different international military conflicts in the middle east. Namely, the Syrian civil war, eastern Ukraine conflict and Afghanistan's war. All images are divided within nine classes, between military vehicles (1–5), environmental situation (6–8) and a final image set (9) featuring other images such as streets and various civilian vehicles: 1) CV 9030 infantry, 2) fighting vehicle, 3) T-72 main battle tanks, 4) Leopard 2A4 main battle tanks, 5) Sisu XA-180 armored personnel carriers, 6) BMP armored personnel carriers, 6) Smoke screen covered tanks, 7) Foliage and camouflage tanks, 8) Tanks firing and 9) Other images.

These images where obtained from diverse social media sources, either as independent shots or as frame extractions from videos [11]. Extracting frames from YouTube videos nearly doubled the volume of data, while also providing images of the vehicles under diverse lighting and weather conditions, and from various angles and distances [11]. Where this data was further labeled to included different vehicles types (Table 1) and situational environments for military machinery (Table 2).

Table 1. Data sources and military vehicles images collected.

	Flickr	Youtube	Web	Total
CV9030	89	262	185	536
Leopard 2A44	124	168	170	462
Sisu XA-180	18	79	143	240
T-72	773	513	171	1457
BMP	182	844	62	1088
Total	1186	1866	731	3783

4 Image Augmentation

Machine learning as an autonomous image classification system has been extensible studied in civilian applications, mainly due the interest of large software

developers and the availability of large image data sets to be trained and tested against [13,17]. Large data bases with thousands or even tens of thousands images per class [13]. Although there is a common practice to test and train against known large data sets, in military applications such images banks does not exist due to security and confidential concerns. Thus hindering the development of neural networks tailored to military applications.

Table 2. Data sources and situational military images collected (including other images class).

	Flickr	Youtube	Web	Total
Smoke screen	0	43	79	122
Foliage and camouflage tanks	0	69	98	167
Tanks Firing	0	11	120	131
Other	819	550	108	1477
Total	819	673	405	1897

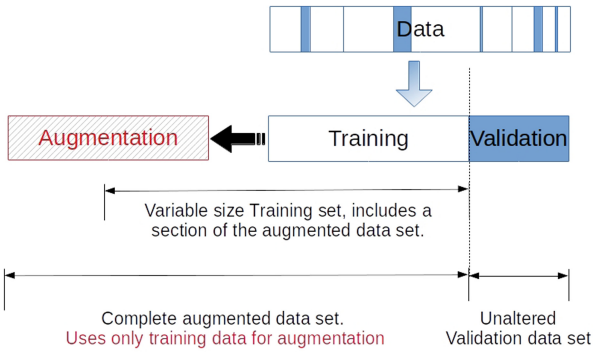


Fig. 1. Data split selection and augmentation set.

In terms of number of images needed to train and validate a convolutional neural network for image classification, the number of images obtained from social media sources fall into the low section of the commonly used. Nevertheless such scenarios have not prevent researchers from finding paths of circumvent issues related with small data sets [11]. As such one approach is to artificially extent the number of images in the data base [26]. To avoid contamination within the validation set and the training data, a special care is made to first shuffle all images in the data base, then split by aleatory selection a 20%/80% portion into Validation/Training data. Further on this artificial image generation is made by exclusively use of the training section of the data, Fig. 1. This can be further

described as an internal loop that freezes all numerical parameters of the neural network training, while selecting continuously a slightly larger section of the training set including an increment part of artificial images added to the system. This method allows for comparison of neural network training structures only in terms of the new augmented data set incorporated into the training. Thus the effect on the type and quality of artificial images can be measured and furthermore optimized.

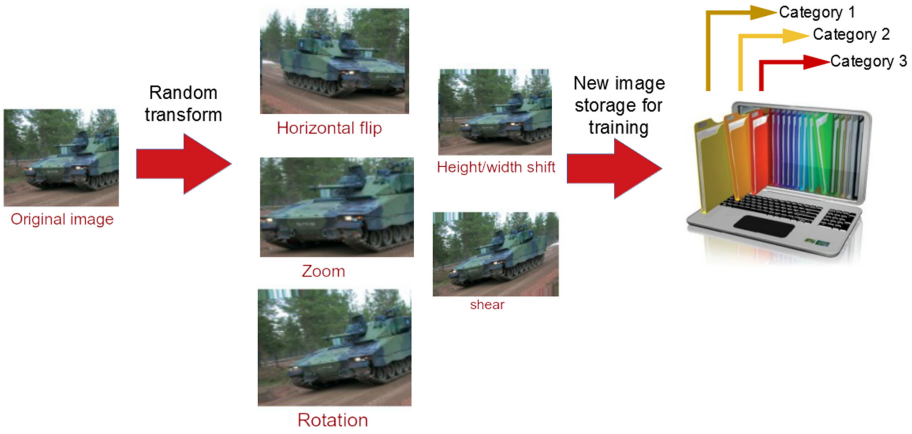


Fig. 2. Image augmentation in practice.

Traditional image augmentation consist of applying random transformations to images in order to generate new training images in the data set [22], that although similar, are sufficiently different for the neural network to learn or refine the necessary features for image recognition. Different type of transformation or filters can be applied to an image to either diversify the data or emphasize certain images features or shapes. Nowadays, there are several means to automatically augment almost every type of data, by either transformations, filters or image concatenations trough neural networks [22]. These techniques have been used successfully in diverse applications, for instance vehicle classification [11], hand writing recognition [26], animal classification [22], among others. Subsequently, proving to increase recognition accuracy and limit the error cause by the neural network recognition, albeit presenting limitations such as artificial images created by random object pasting may hinder accuracy performance [6]. In this investigation the *ImageDataGenerator* class within Keras [2] is used, with different transformation parameters presented in Table 3 and depicted in Fig. 2.

Although this class posses the capability of generate real time data augmentation per batch, a different approach is used on the present research, as from a previously divided training set, a new augmented set is generated with 5000 artificial images. As one of the goals is to evaluate the effect of an augmented

Table 3. Data augmentation: random transformation along with parameters.

Transformation	Parameters
Flip horizontal	True
Rotation	up ± 10°
Shear	max 20°
Zoom range	up to 1 ±0.1
Height shift range	max 5%
Width shift range	max 20%

data set in different neural networks training schemes, this data split and augmentation is previously done once and remains constant for all training cases in the present manuscript. Moreover, these different training schemes are recalculated for diverse size range of augmented images, from 0 augmented images to 5000 in intervals of 1000 augmented images. This is to illustrate the influence of augmentation of data sets in image recognition algorithms depending on how large is the augmented set.

5 Neural Network Training Numerics

Randomness occurs in neural network training schemes, originated from initialization operations, such as weights initial guesses, regularization such as dropout, optimization such as stochastic optimization or other operations presenting aleatory behaviour [2]. Therefore, in order to ensure reproducible results in all training schemes, all random numbers in python numpy [20] and Tensor Flow [1] are seeded by the numbers 1 and 2 respectively for all calculations. Additionally, taking into account previous works in the topic [11], all networks are trained using the hyper-parameters presented in Table 4. All test runs were executed on a physical server with model HP ProLiant DL360 Gen9. The hardware included:- CPU: Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40 GHz with 6 physical cores and 12 threads as frequency from 1.2 to 2.4 GHz. L1d cache 32 KByte, L1i cache 32 KByte, L2 cache 256 KByte, L3 cache 15360 KByte.- 32 GB DDR4 RAM-2 TB HDD.

6 Transfer Learning

Two of the most widely use image recognition networks are ResNet50 [10] and Xception [3], although many others exists [23]. Both of this networks present pretrained weights on ImageNet [13] and are capable to discern images within 1000 classification options (1000 object categories, such as keyboard, mouse, pencil, and many animals, mostly civilian images). The typical approach in cases of small-scale data-sets, (such as the one described above in Sect. 3 of the present manuscript), is to adapt features from existing large trained neural networks onto

Table 4. Neural network hyper-parameters.

Hyper-parameters	Value	Comments
Batch size	64	
Learning rate	1e−3, 1e−4	Decay = 1e−4, 1e−5
Drop out rate	0.2	
Optimizer	Adam	
Image resolution, pixels	224	Optimal for ResNet [2]
Epochs	50	

a new desire target task. To then continue training the network with the available data-set. Transferred neural networks have been already used and adapted to a wide variety of recognition tasks, such as malware recognition using ResNet50 [25] or white blood cell count for disease diagnostics using ResNet50 and Inception [9].

6.1 Transfer Learning Architecture

The architecture on which these networks are based is very different. While ResNet50 is a deep neural network of 50 layers. A compact version of the original 152 layers deep ResNet [10], based on a explicit layer reformulation for learning residual functions in reference with the layer inputs i.e. letting the layers fit a residual mapping. On the counterpart Xception (i.e. Extreme Inception) is a 71 layers deep network developed as an interpretation of Inception, where modules are replaced with depth-wise separable convolutions [3]. Xception is a convolutional neural network where a depth-wise convolution is followed by a point-wise convolution. Of the total amount of layers, 36 are convolutional structured into 14 modules with residual connections [3]. Subsequently, fine-tuned pre-trained deep learning models do not really profit from adding new layers to the system [11,15], as this implies adding a new configuration of layers that does not necessary harmonize with the original architecture philosophy. It is a common task in transfer learning to replace the top layer of a pre-trained neural network with a new top neural network architecture design for the particular new task [9,25]. Thus leaving a new hybrid neural network with a non trainable part, plus new top trainable layers. In the present work, with the objective of no disturb the original optimized architecture of the pre-trained models used, only 3 new dense layers would be added as the new top architecture, distributed as a pyramid configuration of 1024-1024-512 neurons with a drop out rate of 0,2 in the last layer just before the classification *Softmax layer* [1] with the number of nodes equal to the desired classification options.

Moreover, this is taken a step further in the training of the new hybrid neural network, regardless of the pre-trained network used. The new trainable part is optimized as the best classification protocol in terms of model performance for a variable training top layer configuration. In other words, not only the new

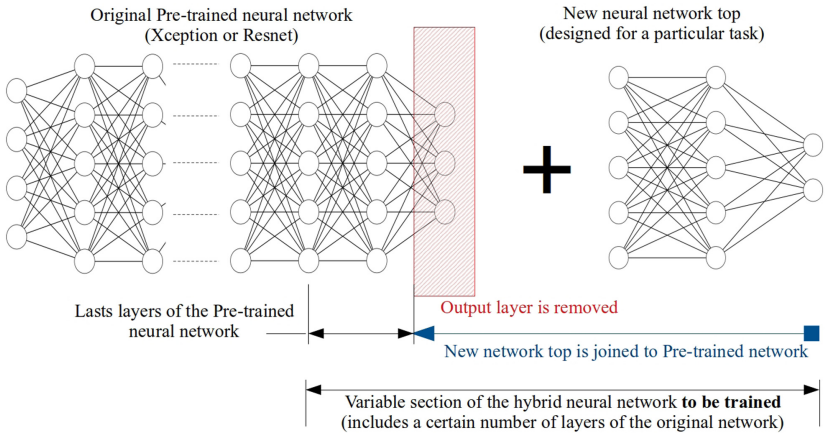


Fig. 3. Transfer Learning architecture.

designed top layers are trained, but also a selected group of layers of the original network using the pre-trained weights as staring conditions for the new training scheme. This is done in a retrospective manner, training different hybrids models with an increasingly training part of the last 5, 10, 15, or 20 layers of the hybrid network, Fig. 3. This training system, not only take advantage of the pre-trained neural network features, but also locates the most suitable trainable layer start that would best adapt this network to the present military application. Further on a cross-validation scheme may be applied to the final transferred network if required.

6.2 Transfer Learning Results

All image classes are separated in five (5) classification cases using both ResNet50 and Xception as base for the transfer learning, Table 5. Where the first three (3) represent image classification in terms of military vehicle type and the final two (2) represent Tanks in different situational environment classification, such as determining if a vehicle is camouflaged or if this vehicle has engage in active fire. All cases present similar behaviour for their respective case and neural network used, therefore only one case would be presented in detail for didactic purposes, CASE: BMP, Leopard 2A4, Sisu XA-180 and Other. Results where oscillating and inconclusive for learning rates values of 1e−3 with decay of 1e−4, after a reduction in these values to 1e−4 and 1e−5 all transferred networks reached stable metrics outlines.

The optimal configuration on number of last trainable layers is selected as the model that provides the highest validation accuracy while still remaining having a relative low Loss value, as seen in Table 5. For the purposes of this paper the Loss function is selected as *sparse categorical crossentropy* [1]. Results for transfer training of ResNet50 and Xception can be seen in Tables 6 and 7 respectively.

Table 5. Optimal training configuration for transfer learning.

CASE	ResNet50: no. last trainable layers	Xception: no. last trainable layers
T-72, BMP and Other	20	15
Leopard 2A4, CV9030 and Other	20	5
BMP, Leopard 2A4, Sisu XA-180 and Other	20	5
Tanks, Foliage and camouflage tanks	15	20
(T72, Leopard 2A4), Smoke screen and Firing Tanks	20	10

To study the stability of the transferred neural networks in terms of validation accuracy and validation loss; all numerical values are filtered with a Savgol filter [21] of window size 9 and polynomial order 3. Furthermore, all approximations are presented with their respective coefficient of determination R^2.

Table 6. Performance on the testing with transfer learning ResNet50.

CASE	Training accuracy	Training loss	Validation accuracy	Validation loss
T-72, BMP and Other	0.99	0.02	0.87	0.94
Leopard 2A4, CV9030 and Other	0.99	0.01	0.90	0.51
BMP, Leopard 2A4, Sisu XA-180 and Other	0.99	0.01	0.95	0.22
Tanks, Foliage and camouflage tanks	0.99	0.01	0.87	0.88
(T72, Leopard 2A4), Smoke screen and Firing Tanks	0.99	0.01	0.81	0.64

In terms of transfer learning for ResNet50, it can be observe that all cases posses not only a high value of validation accuracy between 81%–95% but also low numerical values of validation loss $\leq$0.94. It is important to notice that this represents a well behave neural network only for the optimal number of last trained layers, Table 5. This can be further appreciated in Fig. 4 and 5, where only the optimal configuration (Last 20 Trainable Layers) presents a rather smooth behaviour and the value of R^2 for the filtered function increases as the neural network stabilizes.

In contrast, the transfer learning with Xception presented in Table 7, represents a over-fitted neural network for all levels of trainable layers presented in this research (up to 20), even if this configuration is trained with the same training data and numerical scheme as the previous ResNet50 transfer learning.

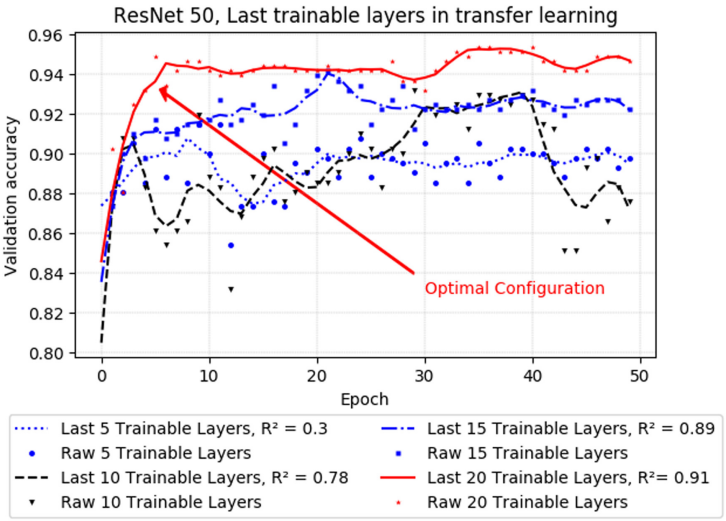


Fig. 4. Transfer learning, ResNet50. CASE: BMP, Leopard 2A4, Sisu XA-180 and Other. Metrics: Accuracy.

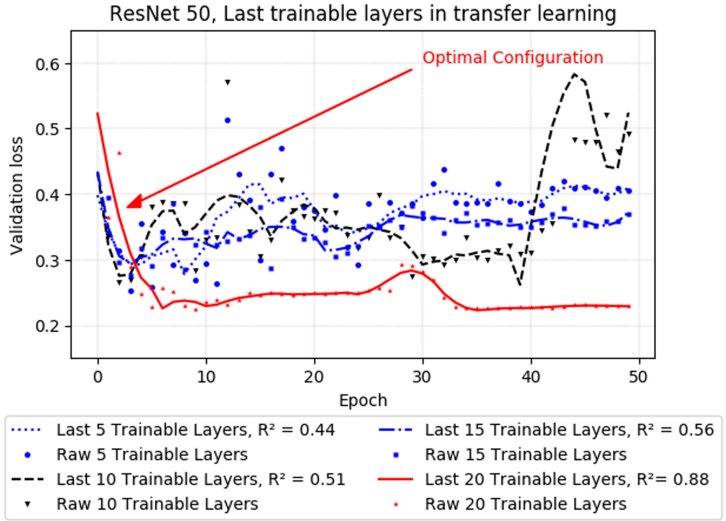


Fig. 5. Transfer learning, ResNet50. CASE: BMP, Leopard 2A4, Sisu XA-180 and Other. Metrics: Loss, sparse categorical crossentropy.

A classical depiction of a over-fitted neural network is present, with a stagnated value of validation accuracy Fig. 6 and a increasing value for validation loss for all configurations with high orders of magnitude in Fig. 7. Not the less, a pseudo optimal configuration or "Best" over-fitted level of training can be selected for further study with data augmentation for every case, Table 5.

Table 7. Performance on the testing with transfer learning Xception.

CASE	Training accuracy	Training loss	Validation accuracy	Validation loss
T-72, BMP and Other	0.99	0.001	0.35	307.8
Leopard 2A4, CV9030 and Other	0.99	0.001	0.36	104.3
BMP, Leopard 2A4, Sisu XA-180 and Other	0.99	0.001	0.27	185.7
Tanks, Foliage and camouflage tanks	0.99	0.001	0.64	9.7
(T72, Leopard 2A4), Smoke screen and Firing Tanks	0.99	0.001	0.39	236.4

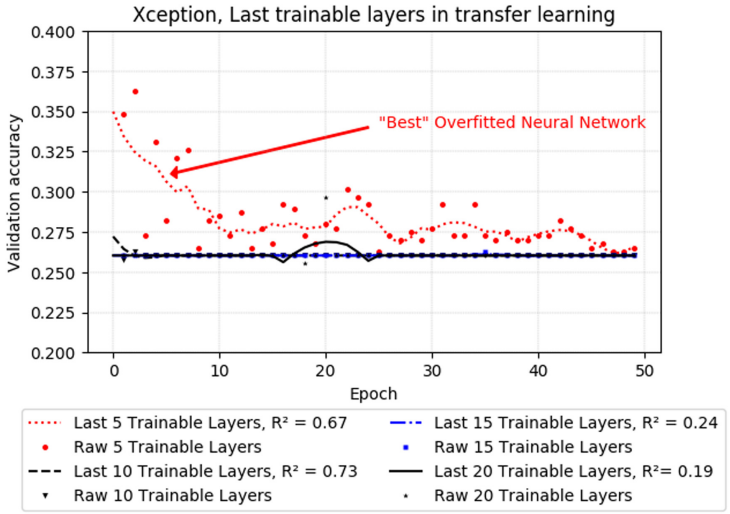


Fig. 6. Transfer learning, Xception. CASE: BMP, Leopard 2A4, Sisu XA-180 and Other. Metrics: Accuracy.

6.3 Effect of Image Augmentation in Transfer Learning

Image augmentation on machinery equipment have prove to be beneficial to image recognition algorithms, although the level of improvement vary depending on the case and algorithm used [26]. In order to measure the level of influence of image augmentation on both trained hybrid neural networks presented, a calculation scheme is build in terms of an incremental number of artificial images feed into the training set. By locking random number generators of the code as previously mentioned, variations on accuracy results can be observe as solely

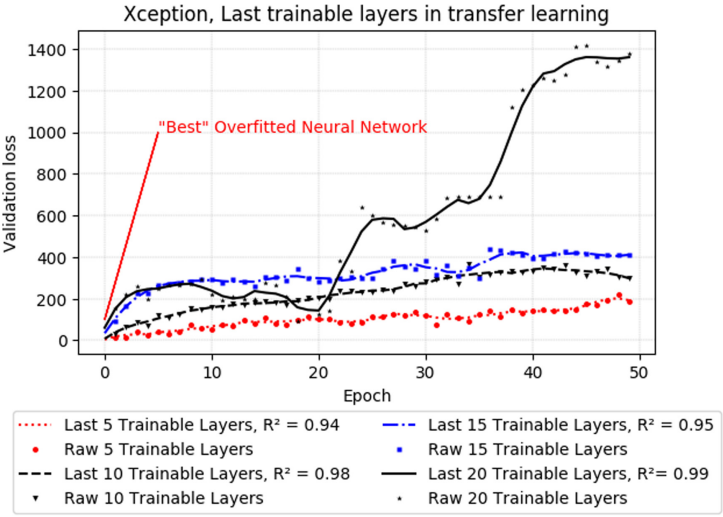


Fig. 7. Transfer learning, Xception. CASE: BMP, Leopard 2A4, Sisu XA-180 and Other. Metrics: Loss, sparse categorical crossentropy.

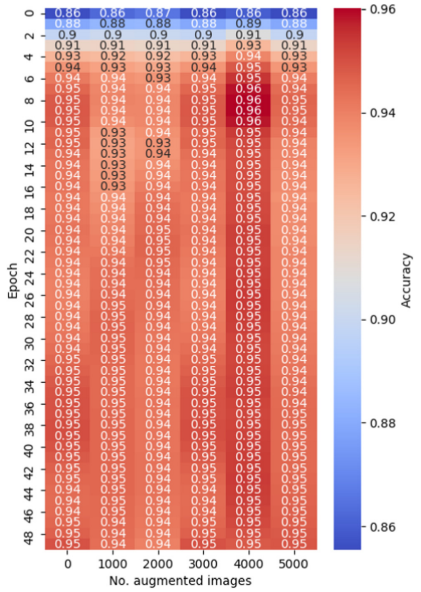


Fig. 8. Transfer learning ResNet50. CASE: BMP, Leopard 2A4, Sisu XA-180 and Other. Effect of image augmentation. Metrics: Accuracy.

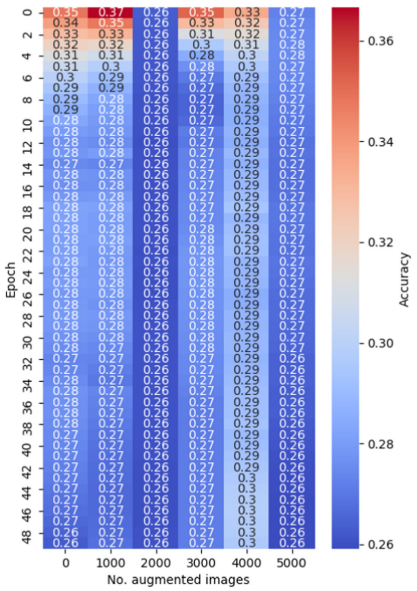


Fig. 9. Transfer learning Xception. CASE: BMP, Leopard 2A4, Sisu XA-180 and Other. Effect of image augmentation. Metrics: Accuracy.

influence of a discrete increment of 1000 fabricated images for several re-runs of the hybrid network.

These results are presented in from of heat maps in Fig. 8 and 9, for the same representative case: BMP, Leopard 2A4, Sisu XA-180 and Other. As it can be observed for the transfer learning of ResNet50, even if there are present small variations at the first training epochs of the transfer; stabilized results towards the end of the training, present no further improvement or drawback. A similar behaviour can be observe for the Xception training, even if it has a lot of room for improvement in terms of accuracy, no significant influence can be observed as function of images feed into the training scheme. Further on, the loss of the system for both hybrid networks either remains unaltered or increases as function of the images feed for all cases studied.

7 Variable Neural Network Architecture

In the present research a slightly different approach from Hiippala (2017) [11] is taken, as it is not assumed that a unique neural network architecture would optimally adapt to all features for all the cases of the present study. As it is hypothesized that different configurations would allocated differently diverse features for recognition in small neural networks, hypothesis that would not necessarily hold true for large neural networks architectures such as optimized and refined cases like ResNet and Xception.

7.1 Deep Learning Architecture

The variable architecture neural network training is automated within specifications delineated in Table 8 and exemplify in Fig. 10. Each own designed architecture is trained from a scratch and tailor fit to the particular case in question for the final classification layer. This variable search algorithm is based on Huttunen (2016) [12] basic neural network for recognition of four vehicles types, with a set of convolutional layers followed by a set of Dense layers to finally enter into the final output layer for classification. In the present case, a variable set of convolutional blocks is couple with a variable set of Dense layers. Where feature maps are flattened after the convolutional layers prior entering the following fully connected layers. Each convolutional layer has an activation function "rectified linear" (ReLU) before a Max-pooling operation. All dense layers are passed trough an activation function ReLU that finally lead to a last Softmax classification layer. To avoid over-fitting all layers have a drop out rate of 0.2 as it has proven to be a reliable value in practice. These variable architecture networks have also a flexible number of neurons that oscillates between 64 and 256 neurons.

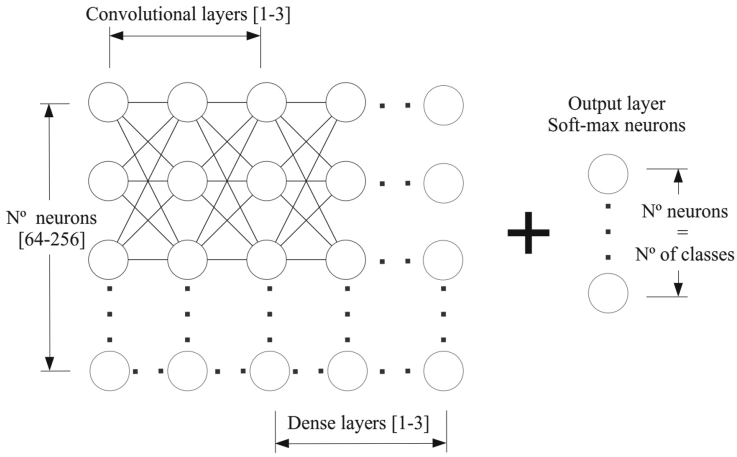


Fig. 10. Neural network architecture.

Table 8. Deep learning neural network architecture.

Neural network element	Optimization search range
Convolution layers	[1-2-3]
Dense layers	[1-2-3]
Number of neurons per layer	[64-128-256]

7.2 Variable Architecture Training Results

Results presented in Table 9 shown that different architectures adapt in a more optimal manner for the different clarification cases. Results prove to be well behaved neural networks for their most optimal configurations, presenting high stabilized validation accuracy results with relative low loss function numerical values. Therefore, in parallel to the transfer learning configuration, only one case would be presented in detail for didactic purposes, CASE: BMP, Leopard 2A4, Sisu XA-180 and Other. Figures 11 and 12 present a few of the tested configurations for the particular case in question. It can be observed that this converged behavior is found within a number of 50 epochs and with a larger learning rate of 1e−3 respective to the transfer learning case. The optimal configuration is selected as the one with the highest stable accuracy value and a relative low Loss function.

Table 9. Optimal Neural Network architecture.

CASE	Neuron no.	Conv. layers	Dense layers
T-72, BMP and Other	64	3	2
Leopard 2A4, CV9030 and Other	64	1	3
BMP, Leopard 2A4, Sisu XA-180 and Other	128	3	2
Tanks, Foliage and camouflage tanks	64	3	1
(T72, Leopard 2A4), Smoke screen and Firing Tanks	256	2	2

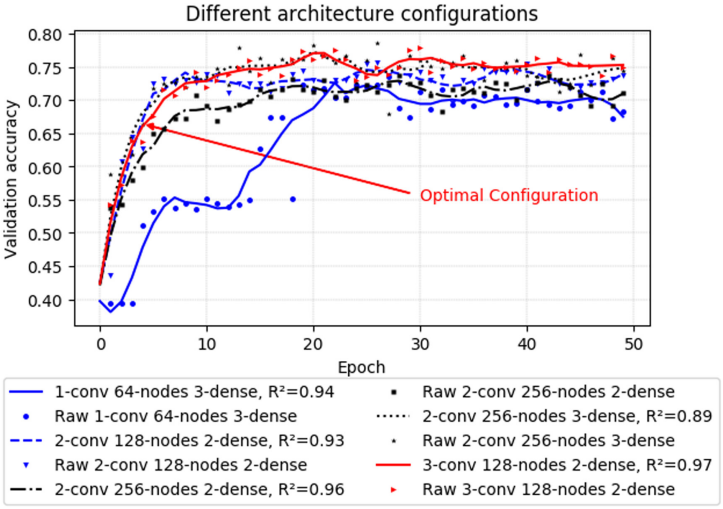


Fig. 11. Own architecture. CASE: BMP, Leopard 2A4, Sisu XA-180 and Other. Metrics: Accuracy.

7.3 Effect of Image Augmentation on Variable Neural Network Architecture Training

Table 10 shows the results for own neural network architecture design with and without image augmentation. It can be appreciated that for the case of training neural networks from scratch, there is an increment in validation accuracy for all cases of ≈10%, with the most accurate neural networks (Val. acc. ≥85%) gaining only a few accuracy points (≤5%) and the least accurate ones (Val. acc. 63%) obtaining an improvement of 20% accuracy due to inclusion of artificial images to the training system.

A detailed influence of how the number of augmented images influence the accuracy of the training process can be observed in Fig. 13 and 14. These represent two different cases in terms of accuracy development. The first case: Tanks,

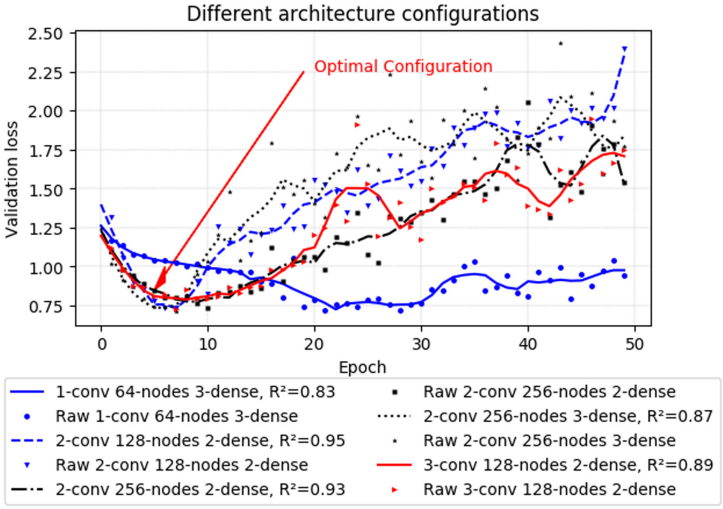


Fig. 12. Own architecture. CASE: BMP, Leopard 2A4, Sisu XA-180 and Other. Metrics: Loss, sparse categorical crossentropy.

Table 10. Performance on the testing with and without data augmentation.

CASE	Validation accuracy	Validation loss	Validation accuracy	Validation loss	Optimal no. aug. img.
	Image augmentation OFF		Image augmentation ON		
T-72, BMP and Other	0.75 ± 0.02	1.3 ± 0.03	0.81 ± 0.01	1.17 ± 0.03	4000–5000
Leopard 2A4, CV9030 and Other	0.87 ± 0.02	0.41 ± .02	0.91 ± 0.03	1.14 ± 0.06	4000–5000
BMP, Leopard 2A4, Sisu XA-180 and Other	0.75 ± 0.02	0.97 ± 0.02	0.85 ± 0.02	0.91 ± 0.04	4000–5000
Tanks, Foliage and camouflage tanks	0.62 ± 0.02	1.34 ± 0.03	0.82 ± 0.02	0.90 ± 0.9	3000–4000
(T72, Leopard 2A4), Smoke screen and Firing Tanks	0.87 ± 0.1	0.62 ± 0.1	0.91 ± 0.01	1.14 ± 0.06	4000–5000

Foliage and camouflage tanks (Fig. 13), portraits how a neural network with a relative low Val. acc. 62% gains large improvements from discrete increments on artificial images, although the increment is not constant and there is a step accuracy decrease to 52% on the first batch of 1000 fabricated images. Nevertheless, reaching its accuracy peak between 3000–4000 augmented images with a top Val. acc. of 83%. The second behaviour can be observe on case: BMP, Leopard 2A4, Sisu XA-180 and Other (Fig. 14); with a rather high Val. acc. (75%). This training system is already stable and shows a slowly but steady accuracy increment towards 4000–5000 artificial images to a sealing value of 86%.

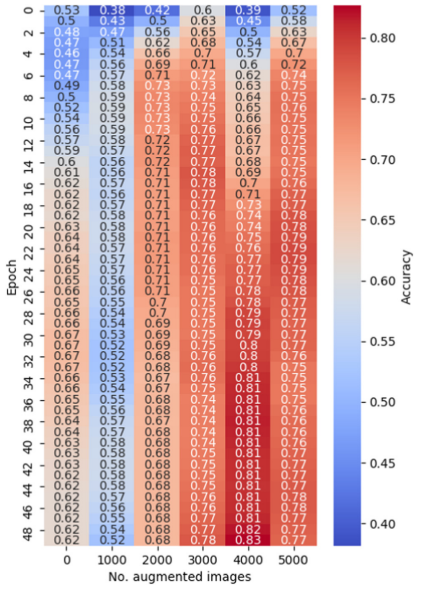


Fig. 13. Effect of image augmentation on neural network training. CASE: Tanks, Foliage and camouflage tanks. Metrics: accuracy.

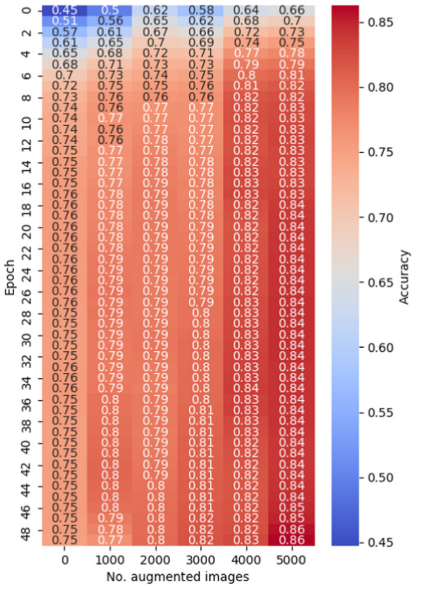


Fig. 14. Effect of image augmentation on neural network training. CASE: BMP, Leopard 2A4, Sisu XA-180 and Other. Metrics: accuracy.

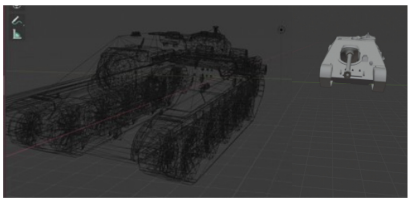
8 Pseudo-random Placements

In terms of machinery equipment since apparatus models have significantly less discrepancies than biological members of a species, 3D designs and advances in rendering techniques have proven to be useful for creating artificial images [19], that in principle provides limitless number of training data for image analysis algorithms. As a future work recommendation; taking into account the limited number of openly available images of military equipment and the secrecy reservations of military institutions in terms of openly sharing large databases of military equipment in use; the following algorithm is proposed as a manner to produce useful artificial images for machine learning algorithms training.

8.1 Algorithm Description

It has been documented that simply allocating images (2D or 3D) on random backgrounds, where the image allocated and the background does not guard any kind of correlation is detrimental for image recognition machine learning training [6]. In consequence some continuity and harmony must be present between the inserted image/3D render and the surroundings. Some developments in this area have been made by artificially inserting 3D ships on true background images for far top air view of ports, shipyards and seashores on flat air view scenarios [27],

accomplishing increasing the quality and quantity of their image training set. Although managing to increase accuracy in their training model, their augmentation algorithm needs further improvement on image overlapping and adequate angles view for the 3D renders, plus adding the necessary model skins to their models for realistic appearance.



A) 3D Military equipment

C) Sphere mapping and level detection

Helsinki (60.164780, 24.949187)

B) 360° Image @ Google [7]

D) Final artificial image

Fig. 15. 3D model placements example.

The algorithm proposed for military vehicles takes advantage of 3D realistic shapes, that in essence can be self drawn/modified with a CAD software from an picture and on top adding camouflage skins and different lighting themes [4] Fig. 15(A). A second step is to obtain from Google maps [7] 360° images through their GPS coordinates Fig. 15(B), these images can be mapped as a sphere projection to set a 360° surrounding, plus adding a plane in tune with the street level of the street view Fig. 15(C). Therefore obtaining a suitable frame to set a military vehicle in this globe surrounding environment Fig. 15(D).

This system has several capabilities such as, various camera angles and flexible zoom focus withing a image background and vehicle pair, diverse illumination configurations, possibility to add fog, rain, fire, snow or sand in the volume around the vehicle plus several backgrounds as street views around the world provided by Google [7] and openly accessible by their GPS coordinates. This algorithm can be added as an extension data set of the augmented images on the different training schemes presented in this paper to measure its influence on the accuracy of different neural networks.

A)
ResNet50:
BMP 100%
Xception:
BMP 100%
Own arch.:
BMP 99.9%

B)
ResNet50:
Leopard 99.9%
Xception:
BMP 99.9%
Own arch.:
BMP 99.1%

C)
ResNet50:
Sisu 96.4%
Xception:
Sisu 85.5%
Own arch.:
Sisu 88.2%

D)
ResNet50:
Sisu 99.9%
Xception:
Sisu 77.7%
Own arch.:
Other 99.4%

Fig. 16. Accuracy predictions. False predictions market in red. CASE: BMP, Leopard 2A4, Sisu XA-180 and Other. (Color figure online)

9 Recognition Examples

In practice the neural networks described in this research produce as an outcome a probability on how accurate is their prediction. Some examples are presented in Fig. 16 for a vehicle type classification and Fig. 17 for a situational environment classification, these images where not used as training in this paper. It is important to notice that results related to the Xception neural network are expected to be the least reliable since this is an over-fitted network with low accuracy ($<64\%$) and high loss ($\gg 1$) values, Table 7. None of the neural networks are capable to make perfect predictions without miss-classifying certain types of vehicles. Although, if the vehicle overall shape can be seen such as in Fig. 16(A),(C), the recognition has higher chances to be on target. On the other hand in 16(B) the own architecture design makes and error, since sharp straight edges and the fact that the tank cannon blend with the pavement lines, makes it somewhat similar to a false prediction of a BMP (canon-less military vehicle). Although this does not mean absolute superiority of the ResNet50 network, even when its Val. acc. is 9% higher than the own architecture design (Tables 6 and 10), there are cases of non military vehicles being miss classified by ResNet50 but correctly allocated by the own architecture design for that particular case, Fig. 16 (D).

The situational environment case: Tanks, Smoke screen and firing tanks is provided as an example in Fig. 17. It is noticed that for ResNet50 Val. acc. $\approx 0.81\%$ and for the case of own architecture design Val. acc. $\approx 0.91\%$, both being quite accurate models. Although, these models work very well to describe the situation of a military vehicle, with errors related to the color of the smoke

or fire. Such as in Fig. 17(C) a miss-prediction is made by ResNet50, since the smoke has certain yellowish tones that produce a false allocation of this image.

It is important to mention that this neural network classification system is in a high degree correlated to the color presented in the image, with yellow/red predominance implying fire and white/gray predominance implying smoke. Nevertheless this case should only be used as a second order recognition system once there is certainty that a military vehicle is located in the figure, since it could produce miss-leading results.

These miss-leading results could be mild, such as mistaking a snow covered civilian car as a smoke cover military vehicle or a military vehicle in a passive state (Basic Tank), Fig. 17 (E). Which in principle does not represent a menace. Or severe errors as miss-labeling a red car with fire-like paintwork as a firing tank, Fig. 17 (D). Since this case would imply an allocated distress situation of a war vehicle engaging in military combat, leading to false information regarding the circumstances of a particular monitored area.

Fig. 17. Accuracy predictions. False predictions market in red. CASE: Basic Tanks (T72, Leopard 2A4), Smoke screen and Firing Tanks. (Color figure online)

10 Conclusions

Two different training schemes of supervised machine learning where studied for classification of military images, variable start layer transfer training models and self design convolutional neural networks training. In terms of transfer learning, ResNet proved to be a well behaved neural network that adapted rapidly to the classification options studied with Val. acc. values on average of 88%. On the other hand, the same transfer-train scheme applied to the Neural network Xception was not successful into adapting to these classification cases, mainly for two reasons, a insufficient large/variable data set and the necessity to train a larger portion of the original network i.e. higher number of last trainable layers.

Moreover, the data augmentation strategy used proved to not have any influence on the accuracy of the transferred machine learning algorithms used. Nevertheless, a new augmentation algorithm is proposed as future work, using 3D renderings of military vehicles in true background 360° images that perhaps could improve the quality and variability of the data to fit pre-trained networks as Xception and many others. The second focus of this investigation is based on variable architecture convolutional network training. Results shown that different architectures are more suitable for different classification cases with average Val. acc. of 86.4%. These are considerable smaller architectures (≈40 times) trained from scratch, and proved to be roughly as reliable as large pre-trained architectures for the purposes of this paper with ≤2% difference.

Acknowledgments. The authors wish to acknowledge Ken Riippa, Jani Haapala and Tuomo Hiippala for labeled data and the CSC – IT Center for Science, Finland, for computational resources.

References

1. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). Software available from tensorflow.org. https://www.tensorflow.org/
2. Chollet, F., et al.: Keras (2015). https://keras.io
3. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp. 1800–1807 (2017). https://doi.org/10.1109/CVPR.2017.195
4. Community, B.O.: Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam (2018). http://www.blender.org. Accessed 09 Mar 2020
5. Bloice, M.D., Stocker, C., Holzinger, A.: Augmentor: an image augmentation library for machine learning. J. Open Source Softw. **2**(19), 432 (2017). https://doi.org/10.21105/joss.00432
6. Dvornik, N., Mairal, J., Schmid, C.: On the importance of visual context for data augmentation in scene understanding, pp. 1–15 (2018). http://arxiv.org/abs/1809.02492
7. Google: Finland google maps. https://www.google.com/maps/@60.1647826,24.9493922,3a,75y,211.07h,75.51t/data=!3m6!1e1!3m4!1swNO3sM2NkZRKrTRN1gqQKg!2e0!7i13312!8i6656

8. Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S.: Deep learning for visual understanding: a review. Neurocomputing **187**, 27–48 (2016)
9. Habibzadeh Motlagh, M., Jannesari, M., Rezaei, Z., Totonchi, M., Baharvand, H.: Automatic white blood cell classification using pre-trained deep learning models: ResNet and inception. In: Tenth International Conference on Machine Vision, Proceedings of SPIE, vol. 1069612, p. 105 (2018). https://doi.org/10.1117/12.2311282
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
11. Hiippala, T.: Recognizing military vehicles in social media images using deep learning. In: IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 60–65 (2017). https://github.com/DigitalGeographyLab/MilVehicles/
12. Huttunen, H., Yancheshmeh, F.S., Ke, C.: Car type recognition with deep neural networks. In: Proceedings of IEEE Intelligent Vehicles Symposium, pp. 1115–1120 (2016). https://doi.org/10.1109/IVS.2016.7535529
13. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. Department of Computer Science, Princeton University, USA (2009)
14. Kaggle: ImageNet Object Localization Challenge — Kaggle. https://www.kaggle.com/c/imagenet-object-localization-challenge/data. Accessed 03 Mar 2020
15. Kornblith, S., Shlens, J., Le, Q.V.: Do better imagenet models transfer better? IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2661–2671 (2019). http://arxiv.org/abs/1805.08974
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
17. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
18. Microsoft: Download Kaggle Cats and Dogs Dataset from Official Microsoft Download Center. https://www.microsoft.com/en-us/download/details.aspx?id=54765. Accessed 03 Mar 2020
19. Movshovitz-Attias, Y., Kanade, T., Sheikh, Y.: How useful is photo-realistic rendering for visual learning? In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 202–217. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_18
20. Oliphant, T.: NumPy: A guide to NumPy. USA: Trelgol Publishing (2006). http://www.numpy.org/. Accessed 09 Mar 2020
21. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
22. Perez, L., Wang, J.: The Effectiveness of Data Augmentation in Image Classification using Deep Learning (2017). http://arxiv.org/abs/1712.04621
23. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016). https://doi.org/10.1109/CVPR.2016.91
24. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement (2018). http://arxiv.org/abs/1804.02767

25. Rezende, E., Ruppert, G., Carvalho, T., Ramos, F., De Geus, P.: Malicious software classification using transfer learning of ResNet-50 deep neural network. In: Proceedings of the 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, pp. 1011–1014 (2017). https://doi.org/10.1109/ICMLA.2017.00-19
26. Sato, I., Nishimura, H., Yokoi, K.: APAC: Augmented PAttern Classification with Neural Networks, May 2015. http://arxiv.org/abs/1505.03229
27. Yan, Y., Tan, Z., Su, N.: A data augmentation strategy based on simulated samples for ship detection in RGB remote sensing images. ISPRS Int. J. Geo-Inf. **8**(6) (2019). https://doi.org/10.3390/ijgi8060276