Data Augmentation and Transfer Learning for Limited Dataset Ship Classification

MARIO MILICEVIC, KRUNOSLAV ZUBRINIC, INES OBRADOVIC, TOMO SJEKAVICA
Department of Electrical Engineering and Computing
University of Dubrovnik
Cira Carica 4, Dubrovnik
CROATIA
mario.milicevic@unidu.hr

Abstract: - Fine-grained classification consists of learning and understanding the subtle details between visually similar classes, which is a difficult task even for a human expert trained in a corresponding scientific field. Similar performances can be achieved by deep learning algorithms, but this requires a great amount of data in the learning phase. Obtaining data samples and manual data labeling can be time-consuming and expensive. This is why it can be difficult to acquire the required amount of data in real conditions in many areas of application, so in the context of a limited dataset it is necessary to use other techniques, such as data augmentation and transfer learning. In this we paper we study the problem of fine-grained ship type classification with a dataset size which does not allow learning network from scratch. We will show that good classification accuracy can be achieved by artificially creating additional learning examples and by using pre-trained models which allow a transfer of knowledge between related source and target domains. In this, the source and target domain can differ in their entirety.

Key-Words: - Deep Learning, Convolutional Neural Networks, Transfer Learning, Data Augmentation, Fine-grained Classification

1 Introduction

Convolutional neural networks (CNN), particularly efficient GPU implementations, are the method of choice for supervised image classification. In CNNs, the convolution has replaced the general matrix multiplication in standard neural networks. Therefore, the number of weights is decreased, reducing the complexity of the network. Furthermore, the images can be directly imported to the network, avoiding the manual feature extraction procedure in standard learning algorithms [1]. Achieving high performance deep learning requires large neural networks models with millions of parameters and large datasets, although larger networks and larger datasets result in longer training times.

Under real-world conditions it is often difficult to get a large number of training samples, and that becomes a handicap to train a deep CNN. As a consequence, the accuracy of classification is reduced and overfitting is a common problem.

Data augmentation is a well-known method for reducing overfitting on models, where the amount of training data is increased using information from training data [2]. Various data augmentation techniques have been applied to specific problems. Current accepted practice for augmenting image data is to perform geometric and photometric augmentations. Geometric transformations alter the geometry of the image with the aim of making the CNN invariant to change in position and orientation. Example transformations include flipping, cropping, scaling and rotating. Photometric transformations amend the color channels with the objective of making the CNN invariant to change in lighting and color [3].

That unsupervised augmentation can also serve as a type of regularization, reducing the chance of overfitting by extracting more general information from the database and passing it to the network [4].

Another noted technique which deals with a limited amount of target training data is transfer learning. While traditional machine learning techniques try to learn each task from scratch, transfer learning techniques try to transfer the knowledge from some previous tasks to a target task when the latter has fewer high-quality training data [5]. In the same paper, authors classified transfer learning into three different settings: inductive transfer learning, transductive transfer learning and unsupervised transfer learning. Furthermore, they classified each of the approaches to transfer learning into four contexts based on “what to transfer” in learning. They include the instance - transfer approach, the feature - representation - transfer approach, the parameter transfer approach and the relational - knowledge - transfer approach, respectively [5].

Donahue et al. [6] show that features extracted from the deep convolutional network trained on large datasets are generic and might serve as very strong features for a variety of object recognition tasks. First layers learn "low-level" features, whereas the latter layers learn semantic or "high-level" features.

The standard transfer learning approach is to train a base network and then copy its first n layers to the first n layers of a target network [7]. The remaining layers of the target network are then randomly initialized and trained toward the target task. The errors can be backpropagated from the new task into the transferred features to fine-tune them to the new task, or layers can be left frozen (they do not change during training the target network).

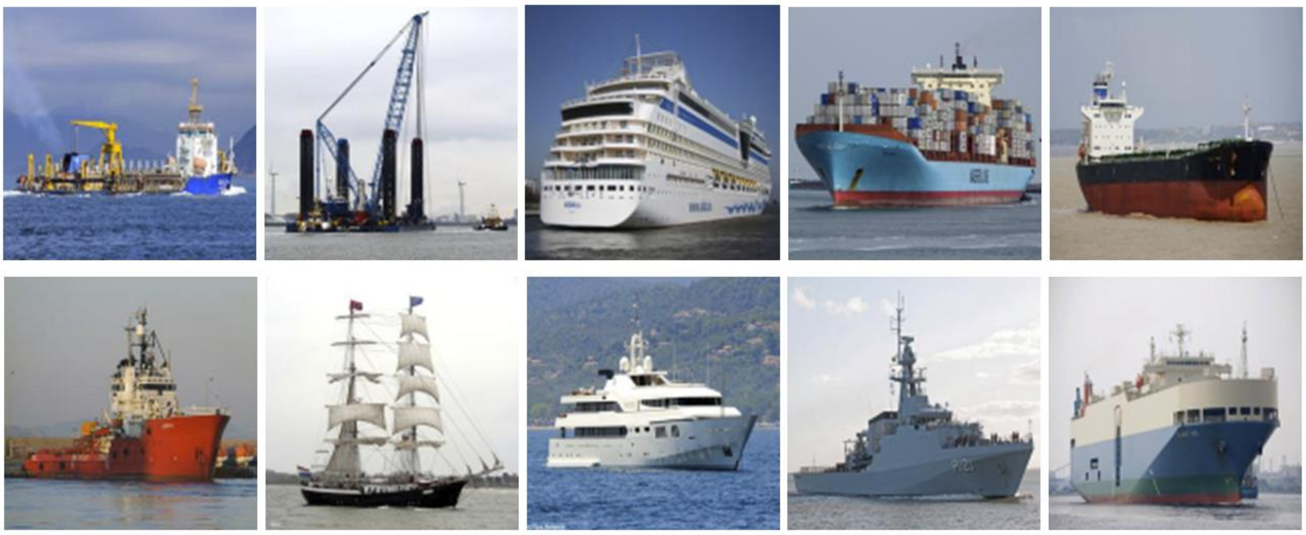

Fig. 1. Examples of vessels from different superclasses

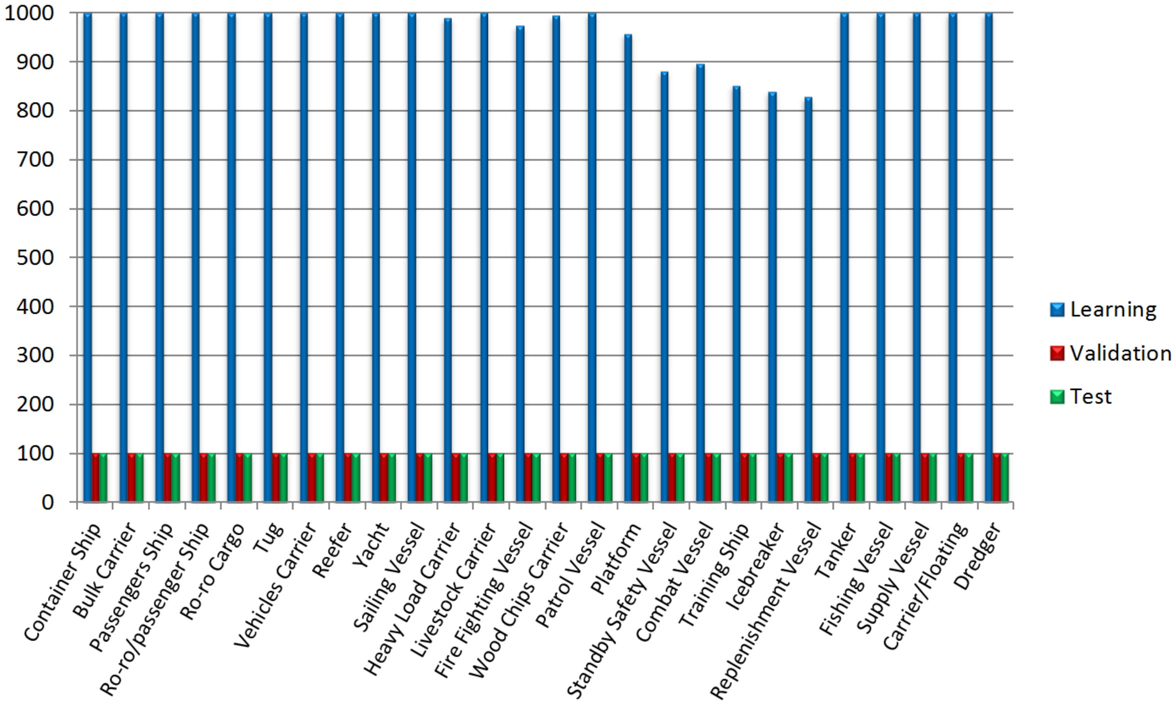

Fig. 2. Distribution of samples in training, validation and test dataset

2 Problem Formulation

CNN training is implemented with the Keras [8] and TensorFlow [9] deep learning framework, using an NVidia GeForce GTX 1080 Ti GPU with 11 GB memory on Ubuntu 16.04 Linux OS.

The dataset used in this study originates from the maritime surveillance systems area. The coastal and marine vision-based surveillance systems containing imaging sensors can also be exploited for the categorization of maritime vessels.

MARVEL dataset [10] originates from 2 million marine vessel images (Fig. 1) collected from the Shipspotting website [11]. Solmaz et al. detect 1,607,190 images with valid annotated type labels belonging to one of 197 vessel categories. By exploiting both the dissimilarity matrix and human supervision, authors merge similar vessel type classes, resulting in final 26 superclasses.

We randomly downloaded only 40,000 images from the Shipspotting website representing a use-case scenario for a limited number of training samples. All images were resized to 256 × 256, and then monochromatic outliers and duplicate images (of the same vessel) were manually removed. The training dataset consists of 25,211 samples, where we tried to acquire equal numbers (1000) of samples from each superclass, but due to the imbalance between superclasses it was impossible to satisfy the requirement of 1000 samples per class (or a total number of 26,000 images).

For this reference dataset, we decided not to generate additional examples by data augmentation, so the classes contain between 828 and 1000 samples - a slight imbalance that will not need particular corrections because data should represent the real-world, where this is a common problem. Both the validation and the test dataset contain 2600 (26 x 100) images (Fig. 2).

3 Results and Discussion

In order to show how far-reaching using data augmentation and transfer learning strategies can be, we conducted an experiment in which we used two different approaches.

3.1 Custom CNN

The first applied architecture (Custom CNN) is a variant of a deep convolutional network (Fig. 3). Input is a fixed-size 256×256 RGB image, with subtracting the mean RGB value as preprocessing step. The image is passed through 4x2 convolutional layers with 3x3 convolutional kernels. The number of output filters in the convolutions is 32, 64, 128 and 256 respectively.

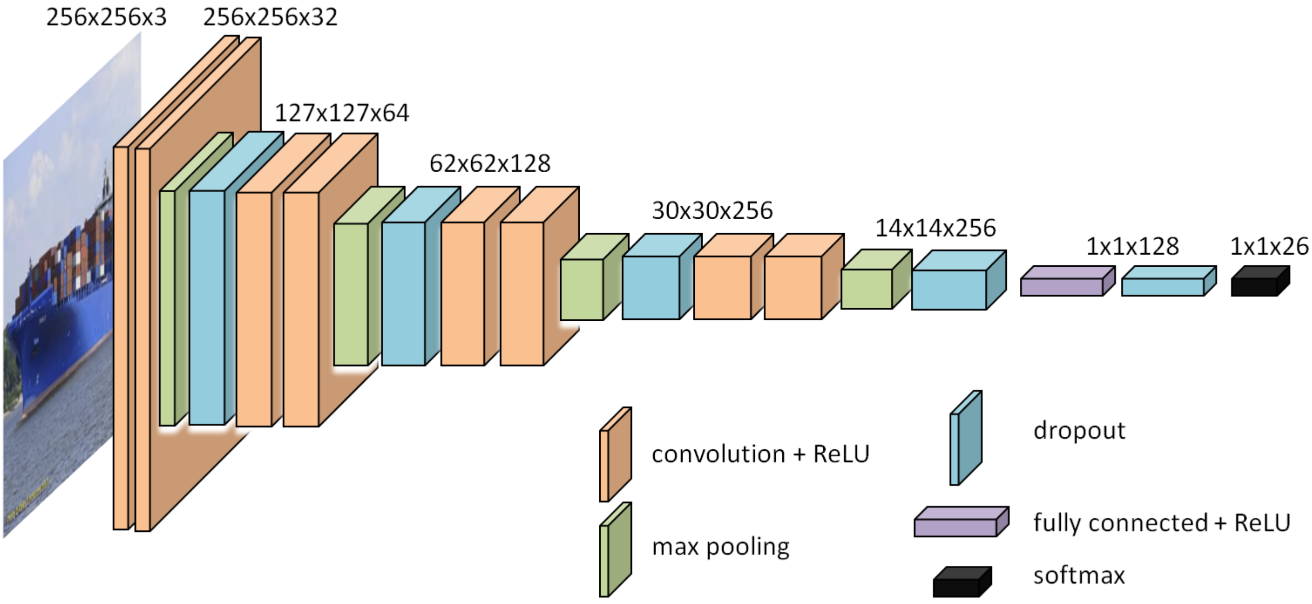

Fig. 3. The proposed initial CNN model architecture

Four max-pooling layers downsample the volume spatially. A stack of convolutional layers is followed by fully-connected (FC) layers, where last one performs a 26-way classification. The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks. All hidden layers are equipped with the rectification (ReLU) [12] non-linearity. This ramp function has better gradient propagation and fewer vanishing gradient problems compared to sigmoidal activation functions. Five dropout layers, with dropout rates between 0.2 and 0.5, are also used to reduce overfitting [13]. In this technique, randomly selected neurons are ignored during training.

Grid search was used for hyperparameters fine-tuning. These are values of some important hyperparameters: number of epochs 200, learning rate 0.00001, mini-batch size 64, RMSProp optimizer [14]. Checkpoint is used to save the best model.

Table 1. Classification accuracies and training epoch duration achieved with different CNNs

Model	Architecture details	Classification accuracy			Epoch duration
		train	valid.	test	
Custom CNN	→ Figure 3 (25211 train. samples)	0.782	0.489	0.497	256 s
VGG19	training from scratch; 25211 train. samples	0.491	0.539	0.525	125 s
	ImageNet weights + train only FC layers; 25211 train. samples	0.986	0.685	0.672	171 s
	ImageNet weights + fine tuning last 5 conv. layers; 5000 train. samples	0.981	0.651	0.649	43 s
	ImageNet weights + fine tuning last 5 conv. layers; 10000 train. samples	0.983	0.729	0.725	72 s
	ImageNet weights + fine tuning last 5 conv. layers; 15000 train. samples	0.987	0.759	0.743	100 s
	ImageNet weights + fine tuning last 5 conv. layers; 25211 train. samples	0.991	0.780	0.762	166 s
	ImageNet weights + fine tuning last 5 conv. layers; data augm. 5000→ 25000	0.998	0.726	0.715	200 s
Inception V3	training from scratch; 25211 train. samples	0.974	0.398	0.376	164 s
	ImageNet weights + fine tuning last 12 conv. layers; 25211 train. samples	0.617	0.335	0.330	65 s
Xception	training from scratch; 25211 train. samples	0.999	0.440	0.432	367 s
	ImageNet weights + fine tuning last 19 conv. layers; 25211 train. samples	0.957	0.541	0.525	128 s
ResNet50	training from scratch; 25211 train. samples	0.978	0.346	0.341	243 s
	ImageNet weights + fine tuning last 22 conv. layers; 25211 train. samples	0.998	0.453	0.460	106 s

3.2 Pretrained Networks and Transfer Learning

This architecture is compared with modern CNN models which achieved state-of-the-art performance on ImageNet [12] and where large pretrained networks can be adapted to specialized tasks. ImageNet is a dataset of over 15 million labeled high-resolution images belonging to roughly 22,000 categories. Visual Recognition Challenge (ILSVRC) uses a subset of ImageNet with roughly 1000 images in each of 1000 categories.

We compare the Keras [8] implementations of VGG19 [15], InceptionV3 [16], Xception [17] and ResNet50 [18]. All these networks can be trained from scratch or initialized with the ImageNet weight for transfer learning approach. The training, validation and test accuracies for these networks are summarized in Table 1.

Among the 1000 ILSVRC classes, only 12 (or 1.2%) of them belong to different vessels (gondola, speedboat, lifeboat, canoe, yawl, catamaran, trimaran, container ship, ocean liner, pirate ship, aircraft carrier, submarine). In this context, it was interesting to analyze how much transfer learning can contribute to the fine-grained vessel classification.

The usual transfer learning approach is to train a base network and then copy its first n layers to the first n layers of a target network. The remaining layers of the target network are then randomly initialized and trained toward the target task [7]. As can be seen in the table, we have experimented with the number of randomly initialized layers. If the target dataset is small and the number of parameters is large, the transferred feature layers can be left frozen, because fine-tuning may result in overfitting.

Practically all models show high accuracy with training data, but, although a pretty aggressive dropout rates is applied, it is not possible to avoid overfitting. The VGG19 network accomplished the best results, which is interesting if it is known that, for example, ResNet50 achieves better results on the ImageNet dataset.

For same training dataset, transfer learning reduces the time span for one epoch at least by a factor of three - compared with learning from scratch. It is also revealed that a decrease in the number of training samples hurts the validation/test accuracy more if the data is already scarce. A drop from 15,000 to 5000 training images (or from 570 to 190 images per class) has a significant impact on the results.

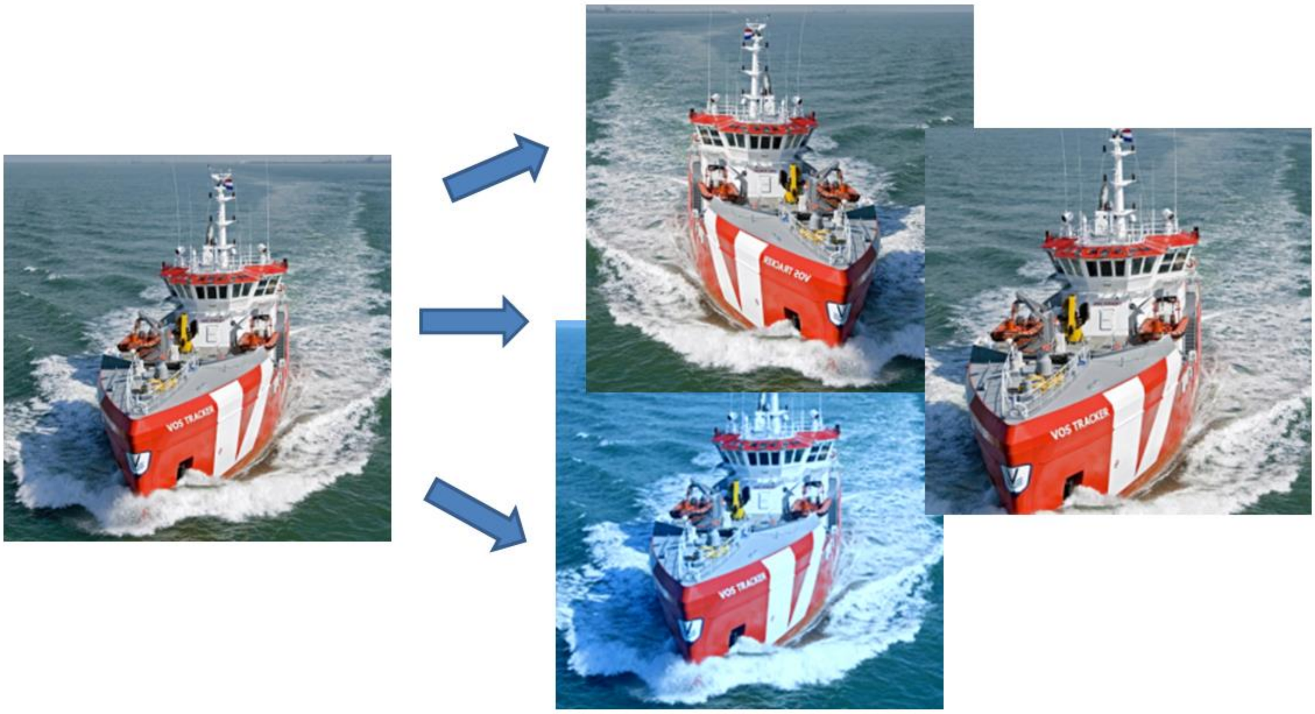

Fig. 4. An augmented image generator

We also evaluate the effectiveness of augmentation techniques, where 25000 augmented images are generated from 5000 source images using horizontal reflections, slight cropping and altering the intensities of the RGB channels (Fig 4.). Without data augmentation, using 5000 training datasets, the VGG19 network achieved accuracy on the test dataset of about 65%. Mentioned data augmentation gives an accuracy of 71.5%, but using training dataset, which has 25,000 original (non-augmented) images, raised the accuracy to 76%.

4 Conclusion

Given the limited number of learning examples, the 78% accuracy achieved for the validation dataset (or 76% for the test dataset) with the VGG19 network and transfer learning is a very good result in the context of 26 classes.

In addition, it should be noted that this is a problem of fine-grained recognition tasks which classify highly similar appearing objects in the same class using local discriminative features. This means that even highly trained human experts sometimes have problems with properly classifying vessel types on a single image.

We also show that data augmentation techniques can be successfully used to benefit fine-grained classification tasks which usually lack sufficient data.

References:

[1] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E.: A survey of deep neural network architectures and their applications. Neurocomputing, 234, 11--26 (2017)

[2] Wang, J., Perez, L.: The effectiveness of data augmentation in image classification using deep learning. Technical report (2017)

[3] Taylor, L., Nitschke, G.S.: Improving Deep Learning using Generic Data Augmentation. CoRR, abs/1708.06020. (2017)

[4] Lemley, J., Bazrafkan, S., Corcoran, P.: Smart Augmentation Learning an Optimal Data Augmentation Strategy, The IEEE Access, Vol. 5: 5858–5869 (2017)

[5] Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22(10), 1345--1359 (2010)

[6] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. Technical report, arXiv preprint arXiv:1310.1531.

[7] Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? Advances in Neural Information Processing Systems 27, pp. 3320--3328, Curran Associates (2014)

[8] Chollet, F.: Deep Learning with Python (1st ed.). Manning Publications Co., Greenwich, CT, USA (2017)

[9] Abadi, M. et al.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, software available from http://www.tensorflow.org (2015)

[10] Solmaz, B., Gundogdu, E., Yücesoy, V., & Koc, A.: Generic and attribute-specific deep representations for maritime vessels. IPSJ Transactions on Computer Vision and Applications, 9, 1--18 (2017)

[11] Ship Photos and Ship Tracker, http://www.shipspotting.com. Accessed 10 May 2018

[12] Krizhevsky, A., Sutskever, I. and Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097--1105 (2012)

[13] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), pp.1929--1958 (2014)

[14] Tieleman, T. and Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning, 4(2), pp.26--31 (2012)

[15] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[16] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z.: Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818--2826 (2016)

[17] Chollet, F.: Xception: Deep learning with depthwise separable convolutions. arXiv preprint, pp.1610-02357 (2017)

[18] He, K., Zhang, X., Ren, S. and Sun, J.: Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770--778 (2016)