The CSI Journal on
Computer Science and Engineering
Vol. 15, No. 1, 2017
Pages 45-51
Regular Paper

Deep Learning for Horse Breed Recognition

Habibollah Agh Atabay

Department of Computer, Gonbad Kavous University, Gonbad Kavous, Iran

Abstract

Recognition of specific types of objects and entities in images has been among the research interests for a long time. In animal species recognition in still images, horse breeds identification has not yet been investigated. In this paper, the identification of horse breeds in natural scene images is addressed. Since a publicly available horse breed dataset of images was not found, a dataset is collected by searching on the Internet. The dataset contains 1693 images of 6 horse breeds. Deep learning methods, specifically Convolutional Neural Networks (CNNs) are dominant approaches today, especially for classifying objects in natural scenes. In this work, Transfer Learning of well-known deep CNN architectures, pre-trained on the ImageNet dataset and fine-tuned on the proposed dataset, is used as the method of choice. Here VGG architectures with 16 and 19 layers, InceptionV3, ResNet50, and Xception are applied as the pre-trained CNN classifiers. The average classification accuracy of ResNet50, the most accurate classifier on the collected dataset, is 95.90 which provides a baseline for further research in the field of horse breed recognition. The results are investigated to specify which horse breed is similar to which, from the perspective of a CNN.

Keywords: Horse Breeds, Deep Learning, Convolutional Neural Networks, Classification.

1. Introduction

Recognition and classification of specific types of objects, such as vehicles [1], faces [2] and plant leaves [3], has been very important research topic in recent decades and has attracted many research attention. Although classification of animal species such as fish [4] and bird [5], and detection of horses in images [6] have been studied in the literature, classification of horse breeds in images, has not yet been considered. If a horse's registration papers are not provided or the animal's parentage is not surely found, educated guessing is needed to determine his breed. If one has familiarity with the basic body types and other characteristics of well-known breeds, he can probably figure out which ones contributed to the horse's ancestry. To do this, various characteristics might be considered, including horse types such as stock types, hot-bloods, warm-bloods, draft, ..., Tattoos and Brands, and DNA features [7]. However, even the most experienced horse people and equine vets might mistake a horse's breed if the horse in question doesn't share obvious traits of his lineage or resembles another breed entirely. In this context, providing an image based tool based on image processing and understanding methods can be very helpful.

The human visual system efficiently classifies images based on the existence of specific objects even within cluttered scenes. But for artificial systems, however, this is still difficult due to the viewpoint-dependent object variability, and the high inter-class variety of many object types. Identification of objects in natural scenes has been a driving motivation for research in computer vision for many years.

The recent advance in object recognition has allowed recognition to scale upwards from a few object instances in controlled setups towards hundreds of target categories in arbitrary environments. This improvement has been gained by the development of robust local image features such as SIFT [8], HOG [9] and deformable part models [10]. Another important factor has been the development of increasingly large and natural image datasets providing various object categories for training and testing, specifically ImageNet [11] and JFT [12].

Bag of Features (BoF) method [8] is one of the widely used approaches to use local features, densely extracted from each image, to aggregate and encode them as high-dimensional vectors, and feed the latter to a classifier, e.g. an SVM. Although this method has been proven to be effective but has some difficulties. Feature Hand-crafting is cumbersome and the devised features often sensitive to the view perspectives and some appearance cues of objects. BoF method discards the spatial information of objects in their histogram representation. Furthermore, the handcrafted features and the capacity of classifiers to differentiate feature information affect the ability of learning algorithms to train the classification models. A horse's body has a non-rigid shape and different sizes that vary greatly among breeds, and therefore it is much more complex to model than rigid objects. Illumination and lighting conditions also affect the image features.

Deep learning methods, specifically Convolutional Neural Networks (CNNs), are dominant approaches today, especially for classifying objects in natural scenes. Using CNNs, which are inspired by the layered structure of mammalian visual cortex [13], feature representations can be learned automatically without the need for hand-engineered features and descriptors. CNNs have recently shown outstanding image classification performance in the large-scale visual recognition challenges (ILSVRC) [14-16]. The success of CNNs is attributed to their ability to learn rich midlevel image representations as opposed to hand-designed low-level features used in other image classification methods. Learning CNNs, however, needs to estimating millions of parameters and often requires a very large number of annotated image samples. Transfer Learning [17] approach allows the weights of a deep network, trained on a large general purpose dataset, to be reused on a small dataset for a specific task. Although the pre-trained model already provides pretty good results on various tasks, these models can be fine-tuned on the desired dataset to further improve the accuracy.

In this paper, deep CNNs with Transfer Learning are used to recognize horse breeds in natural scenes. First, a dataset of horse breeds images is collected from Internet, and well-known CNN architectures, specifically VGGnet [15] with 16 and 19 layers, InceptionV3 [18], ResNet50 [16] and Xception [19], all pre-trained on the ImageNet dataset, are fine-tuned and applied to classify these images. The dataset consists of 6 class of horses each contains between 270 and 300 images, in total 1693 color images, collected using Google search engine.

The training of a deep CNN architecture for horse classification with such a small dataset is prone to overfitting. Two well-known methods to deal with this challenge are data augmentation [20] and Dropout [21] technique applied inside CNN architectures. Here the augmentation method is applied. Because, the images of the dataset are multi-sized, they are simultaneously resized and augmented to use in the CNNs. For augmentation, three cropped versions of each image are extracted (top-middle-bottom or left-middle-right, if the image is wide or tall, respectively). The average classification accuracy of the best model tested on the horse breeds dataset, ResNet50, is 95.90%, which can be used as a baseline for further research.

2. The Dataset

Unfortunately, no publicly available dataset of horse breed images was found for the purpose of this work, thus a dataset is collected using Google search engine. As the first version of the presented dataset, for the challenging problem of horse breed image classification, the breeds were selected such that they were easily classified by the human eye. Six well-recognizable horse breeds are selected, including Akhal-Teke, American Paint, Belgian, Fjord, Shetland pony and Gypsy horses. This dataset can be further extended to include other types of horses and introduce more challenges to the problem. It must be noted that the selected images are just preview

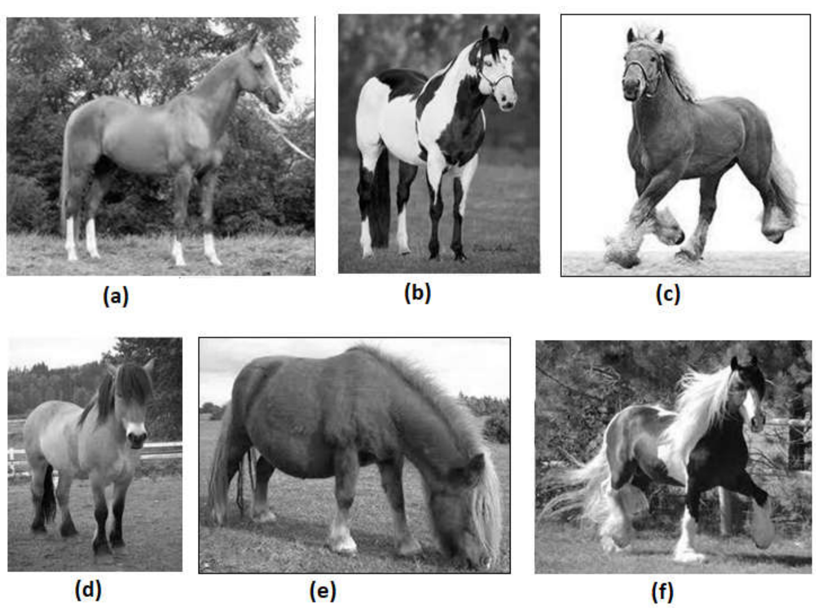

Figure 1: Samples of images in the dataset: (a) Akhal-Teke, (b) American Paint, (c) Belgian, (d) Fjord, (e) Shetland pony and (f) Gypsy.

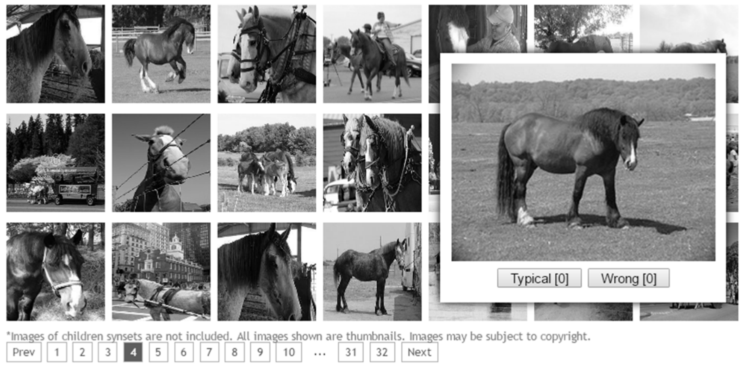

Figure 2: Samples of horse images in Imagenet dataset.

version of the original ones, presented in the result pages of Google, thus they have a rather low resolution. However, high-resolution images in deep learning usually are not needed.

A few physical characteristics of some types of horses, described in [22], are as follows. Akhal-Teke is a slim animal having sparse mane and tail, long muzzle, long and slim neck, narrow chest, long sloped shoulders, and straight legs. In color, Akhal-Teke is often a striking golden dun with a metallic sheen. The American Paint Horse has two major color patterns, Overo and Tobiano. The Overo rarely has white extending across the back between the withers and tail. Head markings are often bald, apron or bonnet-faced. The Tobiano usually has a head marked like a solid-colored horse and may have a blaze, stripe, star or snip. Spots are usually regular and distinct. The Belgian horses are large, powerful horses with willing temperaments. The coat is usually red or chestnut roan, but other colors occur. The head of the Belgian is light and expressive, somewhat squarish in shape. The legs are strong, lean, and sound, with some feather at fetlock. A sample image of each horse type is shown in Fig. 1.

The CNN architecture used in this work has been pre-trained on the ImageNet dataset that includes millions of images from one thousand categories. Among them, there exist some pictures of horses and other similar animals as Fig. 2 shows.

For further research on the dataset used in this work, one can check its download page [23].

3. Transfer Learning of Deep Convolutional Neural Networks

A CNN architecture is a topology that specifies CNN characteristics such as the number of layers and neurons, input and filter sizes etc. Various architectures have been proposed to use in the object recognition tasks, such as LeNet [24], Alexnet [14] and VGGnet [15] and GoogLeNet [25]. These CNNs, except the first one, have been designed for recognition of common object types in a large number of colorful images, and are very deep (having 15 to more than 50 hidden layers, in contrast with “shallow” networks which have only 1 or 2 hidden layers) thus require efficient GPU-based systems to be trained in a reasonable time. The pre-trained models of these architectures on the ImageNet dataset of 1000 object categories, nowadays have been provided by various deep learning frameworks such as Caffe [26], MatConvnet [27], Theano [28], TensorFlow [29] and Keras [30], and their learned information is ready to be transferred and used in other specific tasks such as medical image analysis [31] and plant leaf analysis [32-34].

The process of using pre-trained deep models on a different dataset is called transfer learning [17], where some end parts of architectures are retrained on the new training data. Specifically, when only the fully-connected layers are retrained, the process is called pre-training, and when some convolutional layers (usually the last block of the architecture) below the fully-connected layers are employed, it is called fine-tuning. Transfer learning does not need a huge amount of data and saves a lot of time [17].

For transfer learning, the results of VGGNet [15] with 16 and 19 layers, Inception-v3 [18], ResNet50 [16] and Xception [19] architectures on the horse breed recognition task are compared. VGGNet and the original Inception architecture, GoogleNet [25], yielded similar high performance in the ILSVRC 2014, and ResNet won the first place of the challenge in 2016. The VGGNet architecture consists of 16 and 19 convolutional layers (VGG16 and VGG19) and shows a significant improvement over prior models by using very small convolution filters in the architectures. The original Inception architecture GoogleNet combines the network-in-network approach and the strategy of using a series of filters of different sizes to handle multiple scales. The Inception-v3 is an improved Inception architecture which can be scaled up with high computational efficiency and low parameter count. ResNet is constituted of several stacked residual blocks. Each building block is constructed by a few convolutional layers with an identity connection. It lets each stacked layer fit a residual mapping while skipping connections carry out identity mapping. It is easier to optimize the residual mapping, rather than the original function. The architecture solves the degeneration problem: as stacking more layers, the accuracy gets saturated and then degrades rapidly. ResNet50 is the 50-layer version of the network. Inspired by Inception, the Xception architecture is a stack of depth-wise separable convolution layers with residual connections [16]. The 36 convolutional layers are structured into 14 modules, all of which have linear residual connections around them, except for the first and last modules. Xception is more like InceptionV3 in the number of parameters to train.

4. Experiments and Results

In this section, data preparation, experimental setup and the results of horse breed classification are presented. For the implementations, Keras framework based on TensorFlow backend is used.

Overfitting is a common problem on neural networks, where the overfitted model presents excellent results on the seen data but functions very poorly on unseen test data. A popular method to reduce overfitting in deep learning is data augmentation [20], which is to generate new data samples according to some predefined rules and use them as additive training samples to the original dataset in order to improve generality of the trained model. The most common transformations being used in general computer vision are rotations, scale changes, and random crops. Data augmentation works especially good when applied with deep learning algorithms because the models in deep learning are extremely flexible and are able to learn the representation for the original samples and for the transformed ones. Therefore they are able to generalize also to the variations of the unseen data points [20].

Because of the relatively small sized images of the dataset and the low hardware facilities available for the experiments of this work (a Core i5 CPU with 8 GB RAM), the size of 160×160 pixels is considered for the input layer of all models except for ResNet50 where the condition of the minimum size of 197×197 pixels has been applied to the model, so in this case the size of 200×200 pixels is adopted. To prepare images of the dataset to fit the desired size, the images are simultaneously resized and augmented by extracting 3 cropped versions of each image. To do this, the images are rescaled so that the smaller dimension of them is equal to 160 (or 200), then 3 crops are extracted from top-middle-bottom or left-middle-right, if the image is wide or tall, respectively. Thus the size of the dataset is increased by the factor of 3. The resultant images are further separated by 70%, 10% and 20%, training, validation and test sets, respectively.

The hyper-parameters of the experimented CNNs are set as follows. In all experiments, networks are trained by stochastic gradient descent with the momentum of 0.9 and the learning rate of 1e-4. The mini-batch size is set to 100. The models are trained until the classification accuracy on the training set is reached 99.5% and the best weight set of the model is selected according to the validation results. That is, the model that presents best validation accuracy while training is saved and its classification results on the test set, are presented as the final accuracy of the model.

For transfer learning, the procedure described in [9] is followed. First, the fully-connected layers are removed from the top of the architectures, which are specific to the task and are not reusable. A new randomly initialized fully-connected layers, here the two layers with 1024 and 6 units (the latter corresponds to the number of the target categories) respectively, are replaced. Finally, the deep CNNs are run with stochastic gradient descent (SGD) on the target loss function (Softmax). The Softmax function basically is the multiclass version of the logistic sigmoid function which determines which output of the network is the predicted class for the input. The input layers are also changed according to the mentioned size of images. In pre-training weights of all connections in all layers except the fully-connected layers are frozen and not allowed to update during training. In the case of fine-tuning, some layers below the FC layers are allowed to update. Usually, for each model, the last block in the architecture which consists of a number of convolutional layers is unfrozen. For example, VGG architectures consist of 6 blocks each contains a number of convolutional layers with the filter size of 3×3 and padding 1 and a max-pooling layer with the size of 3×3 and stride 2. The sixth block is commonly chosen for fine-tuning. Similarly, InceptionV3, ResNet50, and Xception consist of 10, 16 and 14 blocks of layers in their architecture. In these architectures also the last block is fine-tuned. The implementation details of fine-tuning with Keras are explained in its author's book [35].

Fine-tuning can be used for any image classification task where a dataset is low-sized. If the class of images is visually different, for example, a few types of distinct flowers, the results of fine-tuning, using same configurations in this work, will be very good in most cases, but if the images are variations of same object or very close in shape, these models will not work easily. The horse breeds dataset probably falls in between of the above categories.

The average classification accuracy, the number of epochs and the approximate time to reach the %99.5 accuracy of training for each fine-tuned CNN architecture, on the horse breeds dataset, are presented in Table 1. In this work, the networks are allowed to gain the same convergence on the training set then the best of them are evaluated to the test set according to the results on the validation set. To calculate the approximate

Table 1: The time and the number of epochs taken to converge on the training set, and the average classification accuracy on the test set, for each model

CNNs	**VGG16**	**VGG19**	**InceptionV3**	**ResNet50**	**Xception**
Number of epochs	125	80	350	405	485
Time needed to train one epoch (Minutes)	21	23	11	35	28
Approximate time (Hours)	43.75	30.66	64.16	236.25	226.33
Average classification accuracy (Percent)	90.69	90.05	88.79	95.90	93

Table 2: The class-specific accuracies of ResNet50

Horse Breed	**Akhal-Teke**	**American Paint Horse**	**Belgian**	**Fjord**	**Gypsy**	**Shetland Pony**
Accuracy (Percent)	97	94	92	96	95	97

Table 3: The most similar class to each one according to the top false predictions of each model

CNNs	Akhal-Teke (AT)	American Paint Horse (APH)	Belgian (B)	Fjord (F)	Gypsy (G)	Shetland pony (SP)
VGG16	APH	AT	F	AT	APH	F
VGG19	APH	AT	F	AT	APH	F
Inception	B	G	SP	AT	APH	B
ResNet50	APH	G	F	AT	APH	F
Xception	APH	AT	AT	AT	APH	B

time for training the time needed to complete one epoch is measured and multiplied by their number as presented in Table 1.

The reason of choosing the threshold of %99.5 for training accuracy, instead of a fixed number of epochs, is that neural networks generally need to reach a good accuracy on the trained set, hoping to approach the same accuracy on the test set (if overfitting does not occur). Some models, like ResNet50, need more epochs to obtain generality and prevent overfitting. So, in this work, the networks are allowed to be trained as long as they reach a high level of training accuracy, and then the accuracy of the resultant model on the test set is calculated as the final score.

The superiority of ResNet50 on classification accuracy is obvious, but it takes a long time to converge and reach to 99.5% accuracy on the training set. Also, the effectiveness of VGG models and their balance of speed and accuracy are considerable. They reach above 99.5% accuracy in less than two days of training and the test results are also desirable. The newly presented Xception model also presented a good accuracy with enough time to converge. The class-specific accuracy of ResNet50 is presented in Table 2. ResNet50 classifies Akhal-Teke and Shetland pony horses better than the others as expected because of their clearly different appearance.

To find out which type of horse is similar to which, from the perspective of a CNN, the top false predictions of the models are monitored. If the prediction of the model about the target class of a test sample (for example, A) is correct, the second prediction (second highly probable class, for example, B) is considered as the similar category, and a vote is assigned to

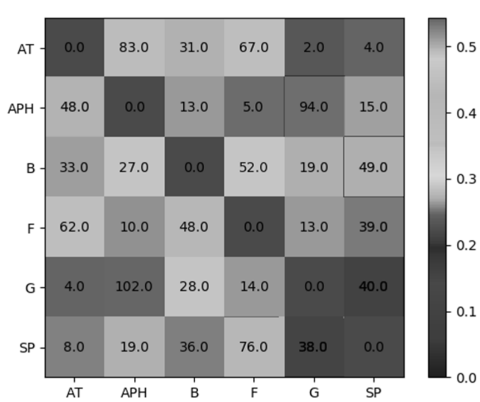

Figure 3: The confusion matrix of ResNet50 for similarity votes. AT, APH, B, F, G and, SP stand for Akhal-Teke, American Paint Horse, Belgian, Fjord, Gypsy and Shetland Pony, respectively.

the similarity of those (A and B). In a case of incorrect classification, the predicted class (the top 1) is considered as the similar category for the target class. The most similar classes for each one, diagnosed by each model, are presented in Table 3.

In Table 3, the similarity of Fjord with Akhal-Teke, and Gypsy with American Paint Horse is confirmed by all models, but for other horse breeds, the votes are distributed. The confusion matrix of the similarity votes of ResNet50 is shown in Fig. 3. Each row shows all votes that assigned to the second probable class. For example in the first row where the target class is Akhal-Teke (AT), in 83 samples of the test set, American Paint Horse (APC) is either the second probable class if the first one is AT (correct classification), or it is the first predicted class where the sample is incorrectly classified. This figure shows that for example, Akhal-Teke is very different from Gypsy and Shetland pony in comparison with others or the American Paint Horse is strongly similar to Gypsy, in the perspective of ResNet50.

5. Conclusions

This paper aimed to provide a starting point in horse breed image classification by gathering a dataset of horse breeds images from the Internet and classifying those using Convolutional Neural Networks (CNNs), the dominant method in object recognition in recent years. The proposed dataset consists of 1693 images of 6 horse breeds, including Akhal-Teke, American Paint, Belgian, Fjord, Shetland pony and Gypsy horses. The deep pre-trained CNN architectures, including VGGs, InceptionV3, ResNet50, and Xception, are applied and their classification accuracy and speed of training are evaluated. It is observed that VGG architectures can be the best choice for training in the limited amount of time and with lower hardware facilities. On the other hand, ResNet50 gives better accuracy but needs a long time to converge. It is also tried to measure the similarity of classes diagnosed by a CNN, using investigation of the top false predictions. The results can be used as a baseline for further research in this field to improve the classification accuracy of the horse breeds on still images and provide better tools for this purpose.

The steps, taken in this work, can be followed for classification of any types of objects in images. The learned features stored in a deep CNN, pre-trained on a large enough dataset like ImageNet, are general enough to recognize distinct variations in objects. With the help of fine-tuning this diagnosis leads to the desired classification results (with some difference in accuracy depending on the problem). But some questions about images may require many steps of reasoning, rather than simply recognizing the main object in the image.

For example, a question like “What color is the thing with the same size as the blue cylinder (as an example)?” is related to how to reason about the problem at hand and needs a new class of models like Neural Module Networks [36]. For one who is interested in this topic, reading the blog post [37] is recommended.

References

[1] X. Wen, L. Shao, Y. Xue, and W. Fang, "A Rapid Learning Algorithm for Vehicle Classification", *Information Sciences*, vol. 295, pp.395-406, 2015.

[2] Y. Xu, Z. Zhang, G. Lu, and J. Yang, "Approximately Symmetrical Face Images for Image Preprocessing in Face Recognition and Sparse Representation Based Classification", *Pattern Recognition*, vol. 54, pp. 68-82, 2016.

[3] Z. Wang, X. Sun, and Y. Zhang, Z. Ying, Y. Ma, "Leaf Recognition Based on PCNN", *Neural Computing and Applications*, vol. 27, pp. 899-908, 2016.

[4] T.T. Hnin, and K.T. Lynn, "Fish Classification Based on Robust Features Selection Using Machine Learning Techniques", *Genetic and Evolutionary Computing*, Springer, pp. 237-245, 2016.

[5] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Bird Species Categorization Using Pose Normalized Deep Convolutional Nets", *arXiv preprint arXiv*:1406.2952, 2014.

[6] M. Uddin, and A. Akhi, "Horse Detection Using Haar Like Features", I*nternational Journal of Computer Theory and Engineering*, vol. 8, pp. 415, 2016.

[7] J. Meggitt, How to Determine a Horse's Breed, Pets, Mom.Me, http://animals.mom.me/how-to-determine-a-horses-breed-8482648.html, November 2017

[8] D.G. Lowe, "Distinctive Image Features From Scale-Invariant Keypoints", *International journal of computer vision*, vol. 60, pp. 91-110, 2004.

[9] N. Dalal, and B. Triggs, "Histograms of Oriented Gradients for Human Detection", *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886-893, 2005

[10] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models", *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, pp. 1627-1645, 2010.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A Large-Scale Hierarchical Image Database", *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp 248-255, 2009.

[12] G. Hinton, O. Vinyals, and J. Dean, "Distilling The Knowledge in a Neural Network", *arXiv preprint arXiv:1503.02531*, 2015.

[13] M. Riesenhuber, and T. Poggio, "Hierarchical Models of Object Recognition in Cortex", *Nature neuroscience*, vol. 2, pp. 1019-1025, 1999.

[14] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks", *Advances in Neural Information Processing Systems*, pp. 1097-1105, 2012.

[15] K. Simonyan, and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 580-587, 2014

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818-2826, 2016

[19] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions", *arXiv preprint arXiv:1610.02357*, 2016.

[20] D.A. Van Dyk, and X.-L. Meng, "The Art of Data Augmentation", *Journal of Computational and Graphical Statistics*, 2012.

[21] G.E. Hinton, N. Srivastava, A. Krizhevsky, and I. Sutskever, R.R. Salakhutdinov, "Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors", *arXiv preprint arXiv:1207.0580*, 2012.

[22] B.L. Hendricks, *International Encyclopedia of Horse Breeds*, University of Oklahoma Press, 2007.

[23] Horse Breeds Dataset, https://goo.gl/MSKc8H, June 2018

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition", *Proc. of the IEEE*, vol. 86, pp. 2278-2324, 1998.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions", *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9, 2015

[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding", P*roc. of the 22nd ACM international conference on Multimedia*, pp. 675-678, 2014

[27] A. Vedaldi, and K. Lenc, "Matconvnet: Convolutional Neural Networks for Matlab", *Proc. of the 23rd ACM international conference on Multimedia*, pp. 689-692, 2015

[28] T.T.D. Team, R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, and A. Belikov, "Theano: A Python Framework for Fast Computation of Mathematical Expressions", *arXiv preprint arXiv:1605.02688*, 2016.

[29] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, and M. Devin, "Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems", *arXiv preprint arXiv:1603.04467*, 2016.

[30] F. Chollet, and Keras, http://keras.io, 2015

[31] D. Shen, G. Wu, and H.-I. Suk, "Deep Learning in Medical Image Analysis", *Annual Review of Biomedical Engineering*, 2017.

[32] G.L. Grinblat, L.C. Uzal, M.G. Larese, and P.M. Granitto, "Deep Learning for Plant Identification Using Vein Morphological Patterns", *Computers and Electronics in Agriculture*, vol. 127, pp. 418-424, 2016.

[33] S.P. Mohanty, D.P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection", *Frontiers in Plant Science*, vol. 7, 2016.

[34] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep Learning for Tomato Diseases: Classification and Symptoms Visualization", *Applied Artificial Intelligence*, pp. 1-17, 2017.
[35] F. Chollet, *Deep learning with python, Manning Publications Co*., 2017.
[36] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural Module Networks", *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 39-48, 2016
[37] J. Andreas, Learning to Reason with Neural Module Networks, BAIR Blog http://bair.berkeley.edu/blog/2017/06/20/learning-to-reason-with-neural-module-networks/, June 2017

Habibollah Agh Atabay received his B.Sc. degree in Computer Software from Zanjan University and M.Sc. degree in Artificial Intelligence from Kharazmi University of Tehran in 2009 and 2011, respectively. His current research areas of interest are Image Understanding, Pattern Recognition and Deep Learning.
Email: atabay@gonbad.ac.ir

Paper Handling Data:

Submitted: 19.03.2018
Received in revised form: 14.06.2018
Accepted: 15.06.2018
Corresponding author: Habibollah Agh Atabay
Affiliation of the corresponding author: Department of Computer, Gonbad Kavous University, Gonbad Kavous, Iran