ArtNet:
An Art Image Classifier fine-tuned from pre-trained Convolutional Neural Networks

Gavin Conran
(gtconran@gmail.com)

Abstract

Using a data set consisting of art images and utilising software development frameworks, a number of art image classifiers, ArtNets, were created by fine-tuning pre-trained convolutional neural networks using the technique of transfer learning. All the pre-trained classifers were, at one time, state of the art and the resulting art classifiers shadow the improved performance of their parent networks. The best ArtNet classifier, based on a combination of Inception v3 and Residual Networks, has a top-1 accuracy of 64.3%.

Dedication

I dedicate this paper to all the broadcast and online educators I have learnt so much from over the last six years. To you I extend my eternal gratitude.

Introduction

The goal of this project was two fold. Firstly, to train and test a model to predict categories of art given an input image. The second had pedagogical ambitions that were, in conducting the project, the author would become familiar with data curation, the best performing Convolutional Neural Networks, the most popular Artificial Intelligence software development frameworks, state-of-the-art image classifiers, the learning transfer technique for fine-tuning pre-existing models, as well as a greater understanding of Western and Japenese art.

The paper starts with the literature review and the charting of the evolution of convolutional neural networks (CNN), a technology that has enabled researchers to achieve impressive performance measures on difficult image classification tasks. The paper then describes the experimental side of the project, by distilling the process of dataset curation and model training. The paper finishes with a discussion of the results, the need for a data science competence in organisations and a cautionary tale of, in the absence of a scientific theory, not to have blind faith in mathematical rules of thumb which are not fully understood.

Literature Review

An in-depth discussion of Deep Learning and Convolutional Neural Networks can be found at [1, 2, 33].

Litjens el al [3] provide a comprehensive overview of Deep Learning for Image Analysis, albeit with a focus on medical imaging, where they take the reader through the main development phases of the technology behind Computer Vision, starting with the “If-Then-Else” based expert systems of the 1990s, followed by the introduction of supervised learning with “handcrafted” feature extraction of the naughties, to today’s “Deep Learning” algorithms, especially Convolutional Neural Networks (CNNs).

Before discussing the evolution of Convolutional Neural Networks it must be mentioned that a contributing factor to the advancement of large scale image classification has been the compilation of large public image repositories, such as Deng et al’s ImageNet [4], and the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) as described by Russakov et al [5], which has acted as a test-bed for large scale image classification systems.

In 1998, Convolutional Neural Networks first raised the bar with the introduction of LeCun et al’s LeNet [6] for character recognition, but it was not until 2012, with the success of Krizhevsky et al’s AlexNet [7] in the ImageNet challenge, that CNNs became the “go to” technology for computer vision. Since then, progress has been swift and there has a been a string of improved models out performing predecessors on the ImageNet challenge, such as Zeiler et al’s ZFNet [8], Zegedy et al’s Inception model [9, 10, 11, 12], Simonyan el al’s VGGNet [13], and Kaiming et al’s ResNet [14, 15].

Training deep neural networks requires large volumes of data, time, and computational resources. In many instances, one or more of these will be missing and researchers will need an alternative method to train a functioning model. Such a method is described by Donahue et al [16], a Learning Transfer technique called DeCAF (Deep Convolutional Activation Features), that allows model designers to reuse, or fine-tune, pre-trained models by adapting to the specific needs of their classification task.

Bahrampour et al [17] conducted a comparative study of a number of Deep Learning Software Frameworks that have emerged to help reseachers quickly build Convolutional Neural Networks. The study was carried out with the following aspects: extensibility, hardware utilisation and speed on five separate frameworks: Theano [21], Torch [22], Caffe [23], Neon [24] , and Google’s TensorFlow [25].

Only two of the afore-mentioned frameworks were used in fine-tuning ArtNets, Jia et al’s Caffe [20] and Google’s TensorFlow, whose programming interface is described by Abadi et al [18] and parallel and distributed capabilities are outlined by Abadi et al [19].

Pre-Trained Networks

The pre-trained ImageNet based networks used to create ArtNet are in ***bold italics*** as shown in Table 1 below. The other networks are included to show the progress and the technical contribution brought to the state-of-the-art.

Net	Contribution	Top 5 Error (%)
2012: AlexNet (Krizhevsky et al)	• Dropout • Data Augmentation • ReLUs • Local Response Normalisation	15.3
2014: ZF-net (Zeiler & Fergus)	• Larger Convolutional Layers	11.7
2014: VGG-net (Simonyan & Zisserman)	• Increased Depth • More Convolutional Layers • 3x3 Convolutional filter	7.32
2015: GoogLeNet / Inception v1 (Szegedy, Liu, et al)	• Novel Inception Architecture • Very Large Network • Dimensionality Reduction	6.67
2015: BN-Inception v2 (Ioffe & Szegedy)	• Batch Normalisation combined with Inception Architecture	4.82
2015: Inception v3 (Szegedy, Vanhoucke et al)	• Factorised Convolutions • Aggressive Dimensionality Reduction	3.58
2015: ResNet50 (He et al)	• Residual Functions integrated in layers	3.57
2016: Inception_ResNet (Szegedy et al)	• Batch Normalisation, Inception Architecture integrated with Residual Functions	3.08

Table 1: CNN Performance on the ImageNet Data Set [26]

Here follows a short description of the networks starting with LeNet, not included in the table as it was not trained on ImageNet.

1998: LeNet

LeNet represents the break from hand-crafted to automatic feature extraction but, due to advances made by more recent CNNs, today, LeNet is used mostly to introduce students to computer vision technology. That said, it is still worthwhile looking at the model as it provides the intellectual foundations of today's state-of-the-art.

The key insight of LeCun et al was the realisation that no learning technique can succeed without a minimal amount of prior knowledge about the task at hand. In the case of multi-layer neural networks, one way to incorporate knowledge was to tailor the architecture to the task. Convolutional Neural Networks are specialised Neural Networks which incorporate knowledge about the variances of 2D shapes by using local connection patterns, and by imposing constraints on the weights.

Armed with this insight, LeCun et al. were able to successfully classify hand written characters with a multi-layer Convolutional Neural Network with the back-propagation algorithm. In addition, they also introduced the concept of Graph Transformer Networks (GTN) which they coupled with Object Oriented Programming (OOP) to form Directed Acyclic Graphs (DAG) which allowed muliple modules, i.e. network layers, to be trained globally to optimise, in this case minimise, a single, global loss function. Back-propagation, DAGs, optimisation algorithms, such as stochastic gradient descent, and loss functions are central to present day Convolutional Neural Networks.

2012: AlexNet

Building on the Convolutional Neural Networks of earlier researchers, Krizhevsky et al added a number of techniques that are still used today. To speed up training they replaced the sigmoid non-linearity neuron with the non-saturating Rectified Linear Unit (ReLU) and utilisied the more powerful Graphical Processing Unit (GPU) rather than rely only on CPUs.

The easiest way to reduce overfitting on image data is to artificially enlarge the dataset by using label preserving transformations such as image translations, horizontal reflections, and patch extractions.

Combining the predictions of many different models is a successful way to reduce test errors, but it appears to be too expensive for big neural networks that already take several days to train. Dropout, however, is an efficient version of model combination that costs about a factor of two during training. The technique of Dropout consists of setting to zero the output of each hidden neuron with probability 0.5. The neurons dropped out in this way do not contribute to the forward pass and do not participate in back propagation. So every time an input is presented, the neural network samples a different architecture, but all these architectures share weights.

On passing the baton to future researchers Krizhevsky et al concluded that improvements could be made by making their network bigger and training it for longer.

2014: ZF Net

Showed how a multi-scale and sliding window approach can be efficiently implemented with a Convolutional Neural Network.

Google Inception

Google's Inception Architecture evolved in stages:

2014: Inception v1 (GoogLeNet)

With their strategic focus on Big Data and Mobile, Google identified the need to improve the utilisation of the computing resources within the convolutional neural network. This was achieved by crafting a network architecture, called Inception [9], that allows for increasing the depth and width of the network while keeping the computational budget constant. For inspiration they turned to nature and Neural Biology's Hebbian principle – Neurons that fire together, are wired together – and the intuition of multi-scale processing.

GoogLeNet is a 22-layer deep model but uses 12x fewer parameters than AlexNet, a 7-layer model, while being significantly more accurate. The main idea of the Inception architecture is based on finding out how an optimal local sparse structure in a convolutional neural network can be approximated and covered by readily available dense components. The second idea is the judicious application of dimension reductions and projections where ever the computational requirements would increase too much otherwise.

A main benefit of the Inception Architecture is that it allows for increasing the number of modules at each stage without any uncontrolled blow-up in computational complexity. An additional benefit is that it aligns with the intuition that visual information should be processed at various scales, similar to freqency capturing in time-frequency analysis, and then aggregating so that the next stage can abstract features from different scales simultaneously.

2015: Batch Normalisation Inception v2

A problem, known as internal covariate shift, with training deep neural networks, is that the distribution of each layer's inputs change during training as the parameters of the previous layers change. This slows down training by requiring lower learning rates and careful parameter initialisation. Szegedy et al [10] over come this problem by making normalisation, i.e. a mean activation of zero and standard deviation of one, as part of the model architecture and performing the normalisation for each training mini-batch.

The benefits of Batch Normalisation include faster training networks, higher learning rates, easier initialisation of weights, more available activation functions, and simplifying the creation of deeper networks, while providing some regularisation.

2015: Inception v3

Google re-emphasies their commitment to Big Data and Mobile in relation to their AI research. Szegedy et al [11] explore the scaling up of networks in ways that aim to utilise the added computation as efficiently as possible by suitably factorised convolutions and aggressive regularisation.

2015: Oxford's Visual Geometry Group (VGG)

Simonyan et al [13] investigated how predictive accuracy would be effected by increasing the depth of convolutional neural neworks. They concluded that a significant improvement can be witnessed by increasing the depth to 16-19 weight layers and stressed the importance of depth in visual representations.

2016: Microsoft Reseach's Residual Networks (ResNets)

Depth improves the accuracy of CNNs but makes training more difficult. In the drive to improve training and by drawing on McCormick's Multi-grid system for solving PDEs [32], He et al [14, 15] introduced the notion of the layers being learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. Their approach led to an empirically proven gain in accuracy with increased depth but with easier optimisation.

2016: Google Inception: Inception ResNet v4

Concludes that there are benefits in combining the Inception architecture with residual connections. Szegedy et al [12] give clear empirical evidence that training with residual connections accelerates the training of Inception networks significantly. There is also some evidence of residual Inception networks outperforming similarly expensive inception networks without residual connections by a thin margin.

Transfer Learning

The models above have been trained with an abundance of labeled data and computing resources but this is not always possible. Donahue et al [16] investigates semi-supervised multi-task learning of deep convolutional representations, where representations are learned on a set of related problems but applied to new tasks which have too few training examples to learn a full deep representation.

The model can be considered as a deep architecture for transfer learning based on a supervised pre-trained phase or simply as a new visual feature DeCAF defined by the convolutional network weights learned on a set of pre-defined object recognition tasks.

Donahue et al validate empirically that a generic visual feature based on convolutional network weights trained on ImageNet out performs a host of conventional visual representations on standard benchmark object recognition tasks. They also found, by visualising semantic clustering properties, that convolutional features appear to cluster semantic topics more readily than conventional features.

Data

Period	**Example**
Renaissance (1300 - 1700) Artists include: • Leonardo da Vinci (1505: Mona Lisa) • Michelangelo • Van Eyck	
Baroque (1600 - 1750) Artists include: • Caravaggio (1608: Crowning of the Thorns) • Rubens • Rembrandt	
Rococo (1750 - 1815) Artists include: • Le Brun (1783: Marie Antoinette) • Watteau • Caneletto	
Neo-Classical (1750 - 1860) Artists include: • David (1786: Oath of the Horatii) • Ingres • Kauffman	
Romantic (1770 - 1850) Artists include: • Turner (1839: The Fighting Temeraire) • Delacroix • Goya	
Realism (1850 - 1870) artists include: • Courbet (1870: Etratet Cliffs) • Manet • MacNeill Whistler	
Ukiyo-e (1620 - 1912) Artists include: • Hokusai (1830: 36 views of Mt. Fuji) • Hiroshige • Utamaro	
Modern (1860 - 1975) Artists include: • Picasso (1932: Girl before a Mirror) • Matisse • Van Gogh	

Table 2: Art Periods (100 images each) included in the Data Set obtained from Google Art [27]

Table 2 above gives an overview of the labeled data set. It has eight categories: Renaissance, Baroque, Rococo, Neo-Classical, Romantic, Realism, Ukiyo-e and Modern. Each category consists of about 100 images. An attempt was made to sub divide Modern into Impressionism, Post-impressionism, Symbolism, Expressionism and Cubic but the performance of the classifier suffered so the decision was made to have a single Modern category. The data set expanded over time and the direction of the growth of the data set was guided by about 40 individual experiments, the results of which producing a direction pivot, using art images and categorisation (labels) from the Google art project [27]. The data exploratory model used for data curation was trained using TensorFlow and the Inception v3 / ImageNet pre-trained model.

ArtNet Learning

As one of the best performing ArtNets was fine-tuned using He et al's ResNet50 /ImageNet pre-trained models with batch normalisation [30] as found in the Caffe Model Zoo [31], for brevity's sake, only its training will be described below.

The data set was randomly divded into trainng (90%) and test (10%) sets. ArtNet was trained with a batch size of 16 and an initial learning rate of 0.1 without any data augmentation. The learning rate followed a linear decay over time. All images were resized such that the smaller side had length of 256 pixels and the aspect ratio was reversed. During training, input images were randomly croped to a 224 x 224 pixel square patch and feed into the network.

Illustration 1 below shows 64 convolutional kernels of size 112 x 112 x 3 of the first convolutional layer of the parent ResNet50 model on the 224 x 224 x 3 input images.

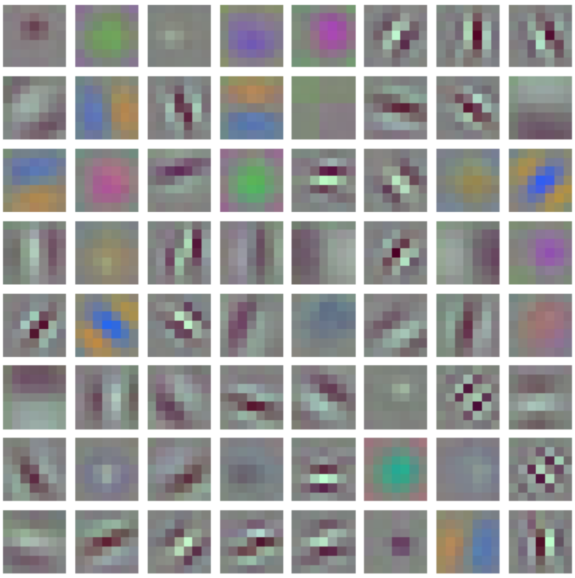

Illustration 1: 64 convolutional kernels of size112×112×3 learned by the first convolutional layer (ResNet50)

As can be observed, the ResNet50 pre-trained model has learned a number of frequency- and orientation-selective kernels, as well as various coloured blobs.

Results

Image	**Prediction Probabilities**
	*1500 Raphael; The Transfiguration* Neoclassical: 29.49% Rococo: 20.43% Baroque: 16.60% **Renaissance: 13.45%**
	*1653 Rembrandt; Aristotle with Homer* **Baroque: 77.39%** Romantic: 12.62% Realism: 3.44%
	*1715 Watteau; Fetes Ventiennes* **Rococo: 33.22%** Romantic: 20.13% Realism: 14.72%
	*1812 David; Napoleon* **Neo-Classical: 70.51%** Rococo: 18.17% Modern: 5.47%
	*1833 Turner; A Disaster at Sea* **Romantic: 65.31%** Modern: 19.57% Rococo: 9.54%
	*1869 Degas; Interior* Baroque: 40.40% Neoclassical: 26.33% **Realism: 20.44%**
	*1857 Hiroshige; 100 Famous Views of Edo* **Ukiyo-e: 99.86%** Modern: 0.04% Renaissance: 0.04%
	*1888 Van Gogh; The Rocks* **Modern: 71.04%** Realism: 11.02% Romantic: 8.34%

Table 3: ArtNet Test Results (fine-tuned on pre-trained Inception v3 ResNet50)

Table 3 is a quantative assessment of ArtNet where there are eight top-3 predictions of images taken from the test set, one for each of the eight categorys. Six of the predictions are correct and one, Degas' Realism painting 'Interior', has the correct prediction in second place. Raphael's Renaissance painting 'The Transfiguration' did not have a correct top-3 classification.

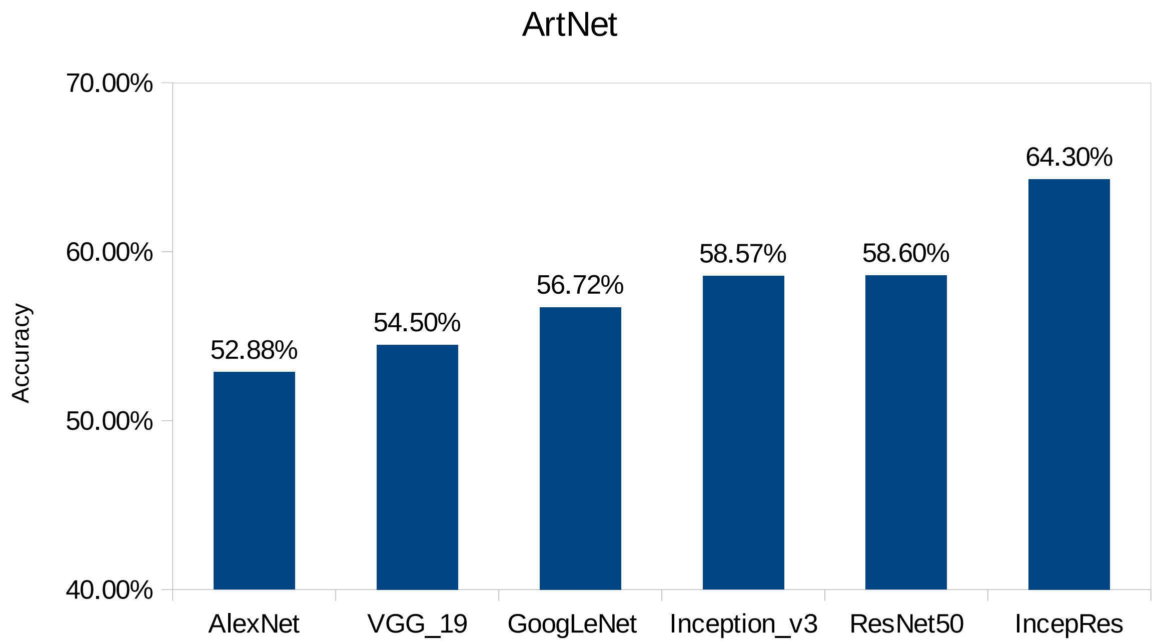

Illustration 2: ArtNet: Model Prediction Accuracy

As shown in Illustration 2 above, the fine-tuned ArtNet classifiers (without data augmentation) shadow the performance improvements of their parent's pre-trained convolutional neural networks, as depicted in Table 1. As mentioned earlier, the best performing ArtNet was fine-tuned using Szegedy et al's Inception v3 / ResNet50 pre-trained model [12], abbreviated to 'IncepRes', as found in the TensorFlow Hub [25]. It is worth noting that data augmentation techniques, such as mirroring, can improve the accuracy of a classifier. For example, training with mirroring improved the accuracy of the ResNet50 ArtNet from 58.6% to 65.8% but greatly increased the training time.

Conclusions

The performance of ArtNet is considerably lower than its ImageNet parent. This is to be expected as art classification by art experts is somewhat subjective resulting in substantial noise in the training and test sets. That said, the ArtNet classifier, on the whole, works well.

As pre-trained nets are off the shelf and software development frameworks, such as Tensorflow and Caffe, abstract away the programming and mathematical details, the critical component to build a successful classifier is data. With the advent of the era of Artificial Intelligence, well curated data sets can become a source of value and should be protected. Management teams should view data strategically and build a data science core competence within their organisations. A well functioning data group can not only improve an organisation's products but also improve internal operations.

The computational mathematical techniques employed by computer vision researchers have been motivated, on the whole, by the desire to reduce the error of the classifier and to speed up model training. With a top-5 error rate of 3.08%, bettering humans, the success of their endeavour is plain to see which leads to the question, what, if anything, can machine learning tell us about human intelligence i.e. is natural intelligence just based on game playing and optimisation, as economists would have us believe or is their a spirtual or instinctive element to it? As Winston [33] demonstrated, humans and machines learn differently, but there may be more in common between natural and artificial intelligence than we realise.

A Cautionary Note

In ‘The Story of Maths’, Marcus du Sautoy [28] introduced the “knotted rope” technique where each knot in the rope represents a corner of a triangle. This invention was used by the ancient Egyptians to help carve out right angled triangles when they were building their pyramids. Du Sautoy observed that the Egyptians had only developed a practical solution, a rule-of-thumb, for right-angled triangles, but did not have a theory to explain them. The world had to wait another 1000 years before Pythagorus [29] shone some light on the subject. Knotted ropes are analogous to trained neural networks. You need a different one for each problem, and, although present day researchers know how to correctly classify an input signal by using learned parameters, i.e. knots, they, like the pyramid builders, really don’t know why their rule-of-thumb solutions work. Which raises a number of questions; is there an equivalent Pythagorem Theorem for Neural Networks, and, are our existing Neural Networks fool-proof? Autonomous cars running over and killing pedestrians would suggest not.

And finally….can a machine have the soul of an artist?

With varying levels of experience humans, like machines, are prone to different levels of error, but is it possible for a machine to feel and think the way a human can when seeing a piece of art, such as Picasso’s Guernica? I think this, for a large part, is what consciouness is and the ability to think and feel, it could be argued, is what differentiates humans from machines.

Illustration 3: Guernica by Pablo Picasso. 1937. Oil on canvas.

But then again, maybe there is nothing mystical or spirtual about emotions, and feelings and intuition are simply optimisation mechanisms that maximise the chances of the survival of an individual human or the species as a whole depending on the circumstance.

Bibliography

[1] Nielsen, M. (2018). Neural Networks and Deep Learning. [online] Available at: http://neuralnetworksanddeeplearning.com

[2] cs231n.github.io. (2018). CS231n Convolutional Neural Networks for Visual Recognition. [online] Available at: https://cs231n.github.io/convolutional-networks

[3] Litjens, G., Kooi, T., Bejnordi, B., Setio, A., Ciompi, F., Ghafoorian, M., van der Laak, J., van Ginneken, B. and Sánchez, C. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, pp.60-88.

[4] Fei-Fei, L., Deng, J. and Li, K. (2010). ImageNet: Constructing a large-scale image database. Journal of Vision, 9(8), pp.1037-1037.

[5] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3), pp.211-252.

[6] Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.

[7] Krizhevsky, A., Sutskever, I. and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), pp.84-90.

[8] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., LeCun, Y. and Fergus, R. (2014). OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks.

[9] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2014). Going deeper with convolutions

[10] Ioffe, S. and Christian, S. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

[11] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2015). Rethinking the Inception Architecture for Computer Vision.

[12] Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A. (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.

[13] Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for large scale image recognition.

[14] He, K. and Zhang, X., Ren, S. and Sun, J. (2015). Deep Residual Learning for Image Recognition.

[15] He, K. and Zhang, X., Ren, S. and Sun, J. (2016). Identity Mappings in Deep Residual Networks.

[16] Donahue, J. and Jia, Y., Hoffman, J., Zhang, N., Tzeng, E. and Darrell, T. (2013). DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition.

[17] Bahrampour, S., Ramakrishnan, R., Schott, L., and Shah, M. (2016). Comparative Study of Deep Learning Software Frameworks.

[18] Abadi, M., Barham, P., Chen, J., Chen, Z.,Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J,. Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. and Zheng, X. (2016). TensorFlow: A System for Large-Scale Machine Learning.

[19] Abadi, M., Barham, P., Chen, J., Chen, Z.,Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J,. Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. and Zheng, X. (2016).TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.

[20] Jia, J., Shelhamer, E., Donahue, J., Karayev, S., Long, L., Girshick, R., Guadarrama, S. and Darrell, T. (2014). Caffe: Convolutional Architecture for Fast Feature Embedding.

[21] Theano. (2018) [online] Available at: http://deeplearning.net/software/theano/

[22] Torch. (2018) [online] Available at: http://torch.ch/

[23] Caffe. (2018) [online] Available at: http://caffe.berkeleyvision.org/

[24] Neon. (2018) [online] Available at: https://software.intel.com/en-us/ai-academy/frameworks/neon

[25] TensorFlow. (2018) [online] Available at: https://www.tensorflow.org/

[26] Rawat, W. and Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review.

[27] Google Arts & Culture. (2018) [online] Available at: https://artsandculture.google.com/category/art-movement

[28] Du Sautoy, M. The Story of Maths (2008) [online] Available at: https://en.wikipedia.org/wiki/The_Story_of_Maths

[29] Pythagorus of Samos. The Pythagorem Theorem (circa. 500 BC) [online] Available at: https://en.wikipedia.org/wiki/Pythagorean_theorem

[30] Simon, M., Rodner, E. and Denzler, J. (2016). ImageNet pre-trained models with batch normalisation

[31] Caffe Model Zoo. (2018) [online] Available at: https://github.com/BVLC/caffe/wiki/Model-Zoo

[32] W. L. Briggs, S. F. McCormick, et al. (2000). A Multigrid Tutorial. Siam,

[33] Winston, P. Neural Networks (2018) [online] Available at: https://www.youtube.com/watch?v=uXt8qF2Zzfo