

# Monument Recognition using Deep Neural Networks

*Siddhant Gada*  
Information Technology  
D.J. Sanghvi College of  
Engineering  
Mumbai, India  
siddhantgada16@gmail.com

*Viraj Mehta*  
Information Technology  
D.J. Sanghvi College of  
Engineering  
Mumbai, India  
viraj.mehta30@gmail.com

*Karan Kanchan*  
Information Technology  
D.J. Sanghvi College of  
Engineering  
Mumbai, India  
karan.kanchan96@gmail.com

*Chahat Jain*  
Information Technology  
D.J. Sanghvi College of  
Engineering  
Mumbai, India  
chahatjain2525@gmail.com

*Prof. Purva Raut*  
Information Technology  
D.J. Sanghvi College of Engineering  
Mumbai, India  
purvapraut@gmail.com

**Abstract**— In this paper, we have classified the famous Indian Monuments across the Golden Quadrilateral. A well-known Deep Learning architectural model has been adopted to provide on-time and striking accuracies for classifying the images. A deep learning library has been used for all the training computations on the classifier for the dataset generated using a web crawler. The concept of Transfer learning has been used to prune the computational load. The last layer of the architecture is retrained as per the training set. Once the model is fully trained, the model is tested on a few arbitrary images to determine the test accuracy of the model.

**Keywords**—Deep Learning, Inception, Landmark recognition, neural networks

## I. INTRODUCTION

The notion of Computer vision, nowadays, is not only circumscribed to bar-code scanning and optical image processing. It is achieving newer heights with advancement in the field of Machine Learning. In recent times, deep neural networks have spurred progress in image recognition. The vital component that led to these results is named convolution neural network [1]. In simple terms, convolution neural network is a special kind of neural network that uses multiple copies of the same neuron and makes a layered architecture of the image weights on which the deep learning algorithm can then be applied. When it comes to landmark recognition from images, the complexity due to their contour and resemblance to other structures has to be taken care of. To suffice this need of attaining higher accuracy and efficiency in complex scenarios, a wide range of images from different angles are to be grouped for a particular landmark and then fed into the database for training.

Convolution neural networks have found their use in a vast spectrum of applications; be it creating highly perceptual artistic images [2], Super resolution of images [3]

to the highly acclaimed and prominent ImageNet classification model suggested by Krizhevsky et al. [4]

The Transfer learning [5] model has been adopted here. Modern object recognition models incorporate a tremendous number of attributes and might take a very long time to train. Transfer learning [5] is a technique that uses a previously trained Neural Network for datasets like ImageNet, MSCOCO, CIFAR-10 etc. and trains it again from the weights obtained after training on such datasets. This methodology will minimize the workload substantially. Though transfer learning is not as effective as training the entire network, it is surprisingly constructive for many cases, and takes significantly less amount of time on a workstation, without involving a GPU.

The first stage evaluates dataset and computes the bottleneck value for each image present in the dataset. The final layer is where the classification is performed and the layer preceding the final layer is used to calculate the bottleneck values. This layer is trained in such a manner that it will give out a group of values which will be capable of differentiating between the various classes of the dataset. It is imperative that there must be a very dense summarization of these pictures, as there exists a very small set of values and the classifier has to make the best choice from what is available to it.

One of the things the script does which is abstracted from the user is that the dataset is first divided into three sets. The training set is generally the largest. In the training phase, each and every image in the dataset is entered into the neural network and the results are used to revise the weights of the training model. All of the images in the actual dataset are not incorporated in the training set. This is done to avoid overfitting. To counter that, some data is kept apart from the training sets after that the model cannot memorize the images. The purpose behind keeping some images apart is so that they can be used to check whether the model is overfitting. For the most part, if high accuracy is seen then it signifies that the

network is not overfitting. The conventional bifurcation is to put eighty percent of the dataset for training, keeping ten percent aside to use for validation frequently during the training, and then have the remaining ten percent that is used for testing to check the performance of the classifying model. The bifurcation can be altered as required. There are also flags dedicated to the percentage value bifurcation of training and testing sets.

After generation of the bottlenecks, we start with training of the uppermost layer of the network. During the training process, after each iteration, there will be a series of outputs each for training and validation as well as the cross entropy. The training quality displays the percentage of images used in the contemporary training batch with the accurate class markings. The validation accuracy is the measure of exactness on an arbitrarily-selected unit of pictures from a varied set. Cross entropy is a loss functional parameter which portrays how well the model's learning ability is increasing. The training's objective is to make the loss as small as possible, so it is possible to figure out whether the model is learning correctly or not by verifying whether the loss keeps trending downwards.

## II. LITERATURE SURVEY

A survey on Landmark Recognition in Deep learning [6] has briefly summarized the previous work done in this field of study which we have incorporated in our paper.

In "Landmark Recognition Using Machine Learning" [7], the training data consists of just 193 images from Google Database, the number being close to the ground. The images are concocted into the same feature dimension. This is done by cropping the images to an aspect ratio of 5:2. The features of the images are extracted using a HOG (Histogram of Oriented Gradients) descriptor. The compelling notion behind the HOG descriptor is that the distribution of the intensity gradients or edge direction scan be potentially described by the local object appearance and shape within an image. The HOG descriptor counts occurrences of gradient orientation in localized portions of an image as well as analyses them which in turn increases the processing time. Also, HOG descriptor is not scale and rotation invariant. The machine learning classifier used is SVM (Support Vector Machine). SVM, being a streamlined supervised learning model, works efficiently for high-definition images. But, when given a large dataset, this models lags. A dataset of 100 images was run on the SVM model which returned an accuracy of 80%. Larger images are trimmed to multiple overlapping cells with identical aspect ratios. Overall, the results are not encouraging showing no promise to detect images beyond the scope of this basic classification algorithm.

In the technical paper of Survey on Mobile Landmark Recognition for Information retrieval [8], the authors determine landmarks with the use of a mobile application. The technology behind this application is Global Feature. Global Feature generalizes the entire pixel-image with single vector and computes with multiple points on the image thus making it more robust. Also, Global Features include contour representations, shape descriptors, and texture features that makes the classification a bit easier for the SVM. Again referring [6], they have used SVM with Color-edged histogram patch (CEHP) to select finest and most suitable images for training. To extract the details of the salient regions of the landmarks and structures, a probabilistic model is used here. SVM is used under discriminative based classification. An SVM classifier is trained for each category of landmarks using one-vs-all strategy.

In Evaluation of Image-Based Landmark Recognition Techniques: [9], the appearance of any monument varies considerably from various angles of observation. Apart from changes owing to different aspects, illumination changes of an object according to the time, external clutters, and changing geometry of the image representation devices are some parameters which affect the change ability of the landmarks. It is pretty tough to apply 3D information when it comes to landmark recognition applications. Hence, it is impossible to use many such object recognition techniques. Standard equalization is used to resample the color values. Certainly, the histogram of color values is bifurcated into 8 assorted classes of approximately equal numbers of pixels. The color image is coded on eight levels using those classes. Deriche's edge detector has been used to calculate the edges and edge elements are coalesced into line segments. Histogram features are computed from the edge image. Five features were evaluated for the same.

## III. PROPOSED ARCHITECTURE

We have opted for the Inception [10] v3 architecture in for classification, shown below in Fig 1. The Inception [10] architecture encompasses a multitude of different layers used to refine the image dataset constantly to obtain a better model. The different layers have been briefly described as follows:

The Convolution layer is the core building block of a Convolutional Neural Network [1] that does most of the computational heavy lifting. This layer has parameters which entail a set of learnable filters. A convolutional layer is embodied by a four-tensor with input channel, output channel, X input position, Y input position being the four dimensions of this tensor. The X and Y input positions are nothing but offsets; because we are essentially "dragging" this tensor across all absolute X and Y positions in the input.

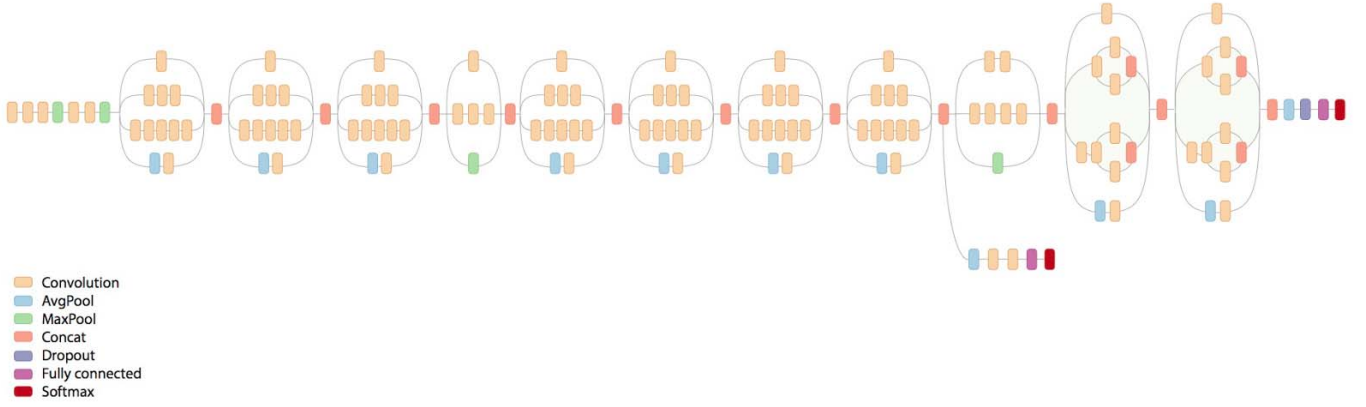


Fig. 1. The inception v3 [11] architecture.

One of the key aspects in successfully training a deep neural network is the activation function. Rectified Linear Unit (ReLU) [12] [13] [14] is the activation function incorporated in this architecture. It is one of the most successful and popular activation functions in CNNs. The mathematical definition of ReLU is  $f(x) = \max(x, 0)$ . Optimization of a network is facile with ReLU as compared to networks with tanh or sigmoid units, because gradients are able to flow when there is positive input to the ReLU function.

Pooling layer is commonly used in CNNs with the aim of gradually reducing the feature space of the image and the computational complexity of the network with max-pooling layer being the most frequently used. Max-pooling is a sample-based discretization process. The main intention is to reduce the input representations and their dimensionality in the process and to make way for assumptions to be made about the features contained in the sub-regions. It reduces the computational cost by reducing the number of parameters to learn.

The fully-connected layers are tasked with the high-level reasoning in the neural network once all the convolutional and max-pooling layers have been taken into consideration. Just like in regular Neural Networks; Neurons, when in a fully connected layer, have complete connections

to all activations in the preceding layer. Since the fully connected layers contain many parameters, it becomes vulnerable to overfitting.

Overfitting is prevented by using the dropout function [15]. The distinct neurons of a particular layer are either "dropped out" of the network with a probability  $1-p$  or they are left untouched with a probability  $p$ , which leaves behind a reduced network. This happens at each training stage. In that stage, only this reduced network is trained on the data. The neurons which were previously removed are inserted in the network again using their original weights.

In the softmax layer as shown in Fig.2, we apply the softmax function on the output of a fully-connected layer (matrix multiplication).

Here, we have an input  $x$  with  $N$  features, and  $T$  likely output classes. The weight matrix  $W$  is used to transform  $x$  into a vector with  $T$  elements (also called logits). The softmax function is used to "collapse" the transformed vector (logits) into a vector of probabilities representing the probability of  $x$  belonging to each one of the  $T$  output classes.

The Concat layer, as the name suggests, is the layer that conjoins its multiple input blobs to one single output blob. It is essentially a utility layer.

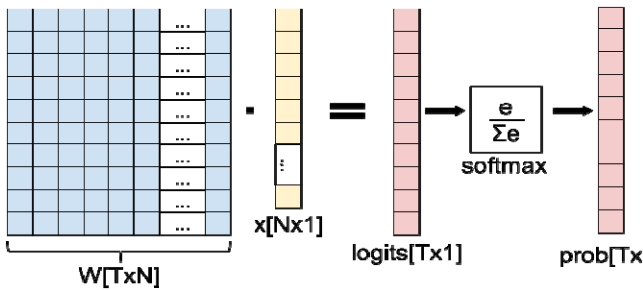


Fig. 2. A generic softmax layer diagram

#### IV. EXPERIMENTAL SETUP

The setup depicts how the final layer is retrained and the computations are performed in the final layer as shown in Fig.3 below. The results of the computations (The training accuracy and Cross entropy measures) have been depicted in the form of scatter plots as shown in Fig.4 and Fig.5.

##### A. Dataset and Preprocessing:

The input dataset consists of the major monuments present in the Golden Quadrilateral of India (Mumbai, Chennai, Kolkata and Delhi). 12 of the most famous monuments in those 4 locations were decided on and then the

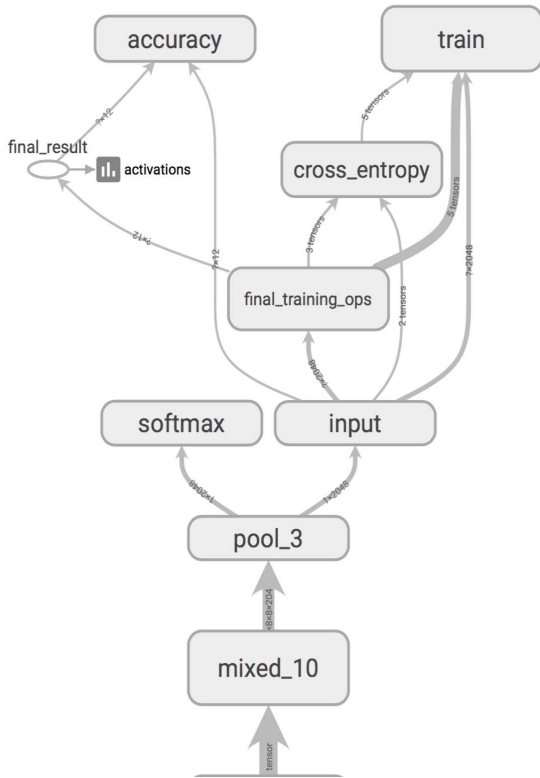


Fig. 3. The final layer of the Inception [10] model

images for each monument were scraped from Google images using a web crawler scripted in Python using the Web Driver API provided by Selenium framework. First, an Instance of the Firefox webdriver is created. The browser.get() method is used wherein the webdriver will open a web browser window containing the “Google images” page of that particular location and the web crawler will start downloading the images. A directory is created for each monument which is scraped and all the images are downloaded into their respective directories. The browser.find\_elements\_by\_xpath() method is used to locate the images within the webpage. Json.loads() method will

retrieve the URL of the image. Using a File object in Python, the image will be first read from the web page and then written into the directory.

Close to 400 images were downloaded for each monument which required some pre-processing in the form of filtering out those images which were irrelevant to the monument in question. This is an important and significant step because such irrelevant images will introduce more noise and distortion into the dataset and will diminish the overall accuracy of the model. For example, If a dataset for a cricket bat is being made, then there might be some images downloaded which would consist of a bat (Mammal). Such images need to be weeded out. Only those images which comprised at the very least, a part which would vividly resemble the corresponding monument were taken into consideration. The next phase was to create bottlenecks for the entire dataset. For each monument, a .txt file is created which will contain the weights of all the images that pertain to that particular monument. Based on the bottlenecks created, the images are then set to train in the final layer of the Inception [10] model.

### B. Retraining the Model

The salient feature of this Transfer learning [5] model as shown in Fig. 3, is that we insert our training set and retrain only the final layer of the Inception [10] model, while leaving the other layers untouched.

This is very advantageous as this gives users the ability to train this model without the requirement of a GPU which is predominantly required when training a dataset consisting of images. Also, the lower layers that have been previously trained can also be reused for multiple other recognition tasks without much modification. There training script will remove the old top layer and it will train a new one on the basis of the training set provided. None of the Monument images will be present in the original class of images on which the entire network was trained on. Once the bottlenecks have been created, the image dataset is set for training.

Cross entropy is a loss functional parameter which portrays how well the model’s learning ability is increasing. The fundamental aim is to minimize the loss as much as possible. After 4000 iterations, cross-entropy value after the

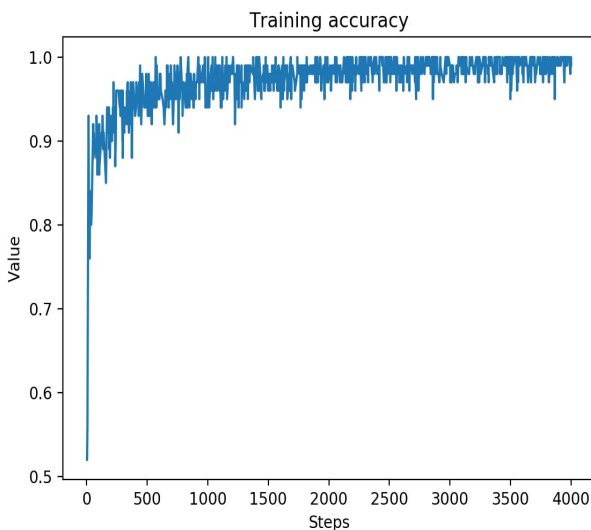


Fig. 4. Training accuracy to the number of iterations

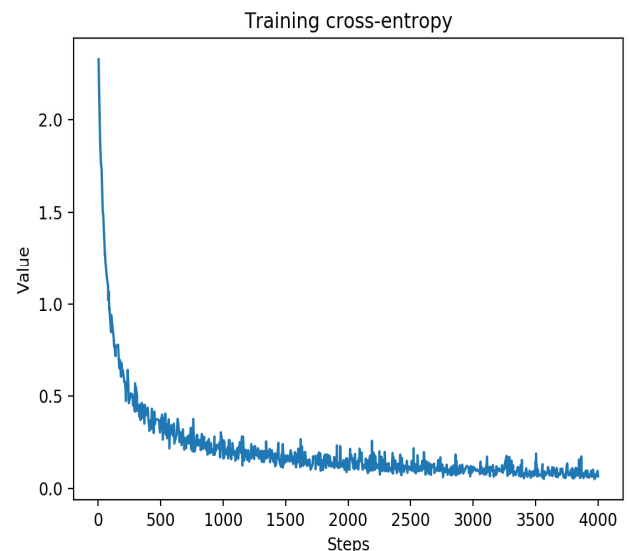


Fig. 5. Cross-entropy to the number of iterations



training is done comes down to approximately 0.067 as shown in Fig 5.

The image data set is trained for 4000 iterations. Alongside that, a labels .txt file is generated which will just list down the labels of all the classes and a graphs.pb file is generated which comprises the final weights of all the images that have been trained on the final layer of the Inception [6] model. Once the dataset has been trained, the model outputs the final training accuracy as 99.4% as depicted in Fig. 4.





## V. RESULTS

Our model provided tremendous training accuracy and minimal entropy measures as shown in Fig.4 and Fig.5 respectively. Once our model was fully trained, we pulled out random images of monuments from the Internet and tested those images against our model as shown in Table I.

In the table, the “Monument to be Classified” column depicts the input image that has been used for testing. The “Location 1”, “Location 2” and “Location 3” columns are used to represent the output values which give the highest score in our model. For example, for image 1, the model outputs India Gate with the highest score and that becomes the value for location 1. The “ground truth” column is the expected output for that particular test image.

We tested our model on 20 such images that were not a part of our training dataset and the accuracy that was obtained was approximately ranging from 96-99% accuracy. Our model was also able to accurately classify grainy images with surprisingly high accuracy. This is a huge step up from the other Machine Learning algorithms in previous models. These results are a testament to the power of the Inception [10] model and the concept of Transfer learning [5].

Table I. Results of the sample test cases

Monument to be Classified	Location 1	Location 2	Location 3	Ground Truth
	India Gate (score= 0.98516)	Gateway Of India (score= 0.01386)	Qutub minar (score= 0.0006)	India Gate
	Howrah Bridge (score= 0.98717)	Worli SeaLink (score= 0.00629)	India Gate (score= 0.00243)	Howrah Bridge
	Victoria Memorial (score= 0.97572)	Red fort (score= 0.00703)	St.Thomas Basilica (score= 0.00576)	Victoria Memorial
	Worli Sea Link (score = 0.93792)	Howrah Bridge (score= 0.02034)	Gateway of India (score= 0.01330)	Worli Sea Link

## VI. FUTURE WORK

Using this deep learning algorithm, the accuracy obtained on this varied data-set is noteworthy. Also it leads us to the fact that different landmarks can be effortlessly recognized using this classification algorithm. Therefore, for further research, we can invariably increase the data-set by adding monuments and structures from all over the globe.

A full-featured smartphone application can be prepared using the algorithm and pre-trained dataset. This application can act as a guide for tourists visiting these monuments or famous architectures. Also if feasible a complete tracker using GoogleMaps and GoogleTranslate APIs can be integrated with this algorithm for engineering a virtual guide that can help the people visiting a place where there are language constraints.

## VII. CONCLUSION

Observing the admirable results with a training accuracy of 99.4% and corresponding testing accuracy ranging from 96-99% and juxtaposing with the other previous models, we conclude that the monuments or landmarks can be accurately identified using this classification model. Using the concept of Transfer learning [5] on the Inception [10] v3 architecture, our final model achieves commendable performance on esoteric visual recognition objects and possesses the ability to classify monuments who have striking resemblance in their appearance with considerate accuracy. The entire model was trained and tested on an Intel 2.7GHz quad- core i7 Skylake processor and it took approximately 30 minutes to train the entire dataset consisting of around 2600 images on the Inception[6] model. This implies that there is an added advantage to using this model that being any user without access to a Graphic Processing Unit (GPU) can also train their dataset of a relatively substantial size within a considerable frame of time. Finally, this model grants the potential for classification of monuments to almost work at par with human vision and ability to perceive them.

## VIII. REFERENCES

- [1] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D., “Backpropagation applied to handwritten zip code recognition.”, Neural Comput., 1(4):541–551, 1989
- [2] Gatys, L.A., Ecker, A.S., Bethge, M., “A neural algorithm of artistic style,” arXiv preprint arXiv:1508.06576 (2015)
- [3] J. Kim, J. K. Lee, and K. M. Lee., “Deeply-recursive convolutional network for image super-resolution.”, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016
- [4] Krizhevsky, A., Sutskever, I., and Hinton, G. E., “ImageNet classification with deep convolutional neural networks.”, In NIPS, pp. 1106–1114, 2012

- [5] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2013), "Decaf: A deep convolutional activation feature for generic visual recognition.", CoRR, abs/1310.1531.
- [6] Sabir A. Kazi, Kshitij A. Bulkunde, Abhik Chakraborty, Kajal P. Dhumal, Dr. Kishor Wagh, "A Survey on Landmark Recognition with Deep Learning", International Journal of Innovative Research in Computer and Communication Engineering (IJRCCE), Vol. 4, Issue 10, pages 17378-17384.
- [7] A. Crudge, W. Thomas and K. Zhu, "Landmark Recognition Using Machine Learning," CS229, Project 2014.
- [8] Tao Chen, Kui Wu, Kim-Hui Yap, Zhen Li, and Flora S. Tsai, "A Survey on Mobile Landmark Recognition for Information Retrieval", IEEE Pp.978-0-7695-3650-7/09, 2009.
- [9] Takeuchi, Y., M. Hebert, "Evaluation of Image-Based Landmark Recognition Technique," CMU-RI-TR-98-20, Carnegie Mellon, 1998.
- [10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna., "Rethinking the Inception Architecture for Computer Vision.", ArXiv e-prints, December 2015.
- [11] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems", arXiv:1603.04467, 2016.
- [12] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung., "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit.", Nature, 405(6789): 947, 2000.
- [13] Kevin Jarrett, Koray Kavukcuoglu, Yann LeCun, et al. , "What is the best multi-stage architecture for object recognition?", In 2009 IEEE 12th International Conference on Computer Vision, 2009.
- [14] Vinod Nair and Geoffrey E Hinton., "Rectified linear units improve restricted boltzmann machines.", In International Conference on Machine Learning, 2010.
- [15] Nitish Srivastava , Geoffrey Hinton , Alex Krizhevsky , Ilya Sutskever , Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.", The Journal of Machine Learning Research, v.15 n.1, p.1929-1958, January 2014