A SURVEY ON NEURAL NETWORK MODELS FOR CLASSIFICATION OF PHENOMENA OF INTEREST IN IMAGES

ABHAY NIKUMBH [1] ,TANMAY RASTOGI [2], MEHUL INGALE [3], SAGAR VARMA [4]
PRIYA SURANA[5]

[1] 1nikumbhabhi2@gmail.com, [2] tanmayrastogi9@gmail.com, [3] mehul.ingale@gmail.com,
[4]sagarvarma.2703@gmail.com, [5] priyasurana1980@gmail.com

[1,2,3,4] - Student, Department of Computer Science Engineering, Pimpri Chinchwad College of Engineering, Pune, India
[5] - Asst. Professor, Department of Computer Science Engineering, Pimpri Chinchwad College of Engineering, Pune, India

ABSTRACT—Forests cover 31 percent of the world's land surface, just over 4 billion hectares. Due to various problems, geographical as well as social, the development of various forested areas is still not achieved. Some of the reasons are:-lack of education, lack of connectivity between places, lack of reliable government policies and the most important one is that there is very little or even no record of various places in the books. The main reason for this is the non-accessibility of various locations of the forest areas due to numerous issues. Hence in this paper we will provide a survey of ways to classify various geographical and environmental phenomena of interest such as haze/fog, land, river, forest fires, etc. In this paper we will describe various data processing techniques like augmentation in detail and describe various neural network architectures best suited for solving these types of problems in detail with different convolutional neural networks (CNN).

KEYWORDS—CNN, classification, transfer learning, VGG, ResNeXt, AlexNet, Xception.

1. INTRODUCTION

The forests in the world cover around 4 billion hectares of land. Forests are the most biologically-diverse ecosystems on land, home to more than 80 percent of the terrestrial species of animals, plants and insects. They also provide shelter, jobs and security for forest-dependent communities.In Spite of this,the world's natural forests and other critical ecosystems like grasslands are hanging on by a thread. Half of the world's forests have already disappeared, and only 20 percent of what remains is intact. As it stands, the world loses more than 23 million acres of forest area every year. The window of opportunity to reverse deforestation and protect the world's remaining intact forests is shrinking very fast. Not only does this have huge consequences for the climate and for wildlife, but it's also a major human rights concern. Some of the main causes of this issue are:- Agribusiness, illegal logging, illegal mining, forest fires, industrial development, etc.

This paper focuses on a survey and analysis of an approach for solving the classification problem of various natural and geographical elements: Transfer learning with CNN. Need for this survey is vital because classification of geographical phenomena is essential for gaining knowledge about various forests, their nearby areas which will help in many industrial and government sectors in development and protection of these areas. Nowadays huge amounts of data is very easily available everywhere, but its raw data and hence no proper insight can be obtained from it.Data needs appropriate rectification. To deal with such a huge amount of data could be very time consuming using traditional feature based methods(Machine Learning).That's why deep learning techniques, especially CNNs are widely used for classification of images.

This paper presents a broad survey on CNN based classification techniques. CNNs are proven very robust towards classification of natural elements. We have done a survey on this, addressing all major problems and their available solutions like pre-processing and data augmentation. The paper discusses the mentioned challenges and highlights how those challenges are tackled by existing works. We have also shown the usage of CNN based techniques in the paper. Moreover, characteristics of CNN architecture are also analyzed to suggest an approximate architecture as per the characteristics of the dataset.

2. SECTIONS

The set of paper is as mentioned below. Section 3, describes the Background knowledge of CNN layers. The preprocessing method of image augmentation is described in Section 4. Section 5 says about the CNN loss functions and 6 about various optimization techniques. Section 7 describes the architectures of common CNN. Section 8 describes the dataset information. Section 9 tells about the implementation.

3. BACKGROUND KNOWLEDGE

A convolutional neural network (CNN) is a specific type of artificial neural network that uses perceptrons, a machine learning unit algorithm, for supervised learning, to analyze data.The structure of CNN is as follows. It can be visualized as a collection of neurons arranged as an acyclic graph. The deep architecture of the network results in hierarchical feature extraction i.e. the trained filters of the first layer can be visualized as set of edges or color blobs, of the second layer as some shapes, the next layer filters might learn object parts and the filters of final layers can identify the objects.A covnets is a sequence of layers, and every layer transforms one volume to another through differentiable function.

Types of Layers:-

1. Input Layer:

This layer contains raw images in terms of the matrix of RGB.

2. Convolution Layer:

This layer computes the output volume by computing dot product between all filters and image matrix.

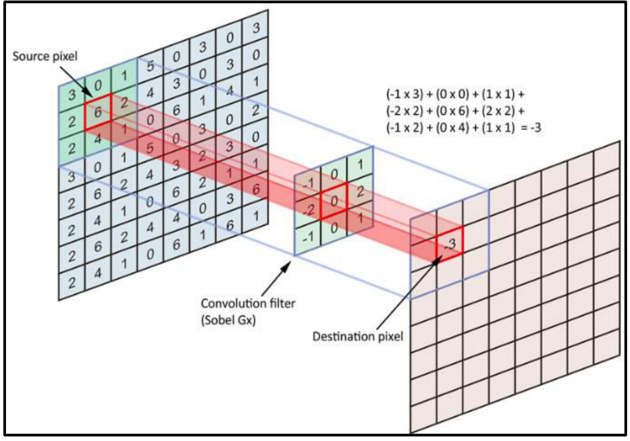
Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
Convolution filter
(Sobel Gx)
Destination pixel

Fig.1: A filter and working of Convolution layer.

3. Activation Function Layer

This layer will apply element wise activation function to the output of the convolution layer. Some common activation functions are:

- RELU: max(0, x)
- Sigmoid: 1 / (1+e^{-x})
- Tanh, Leaky RELU, etc.

4. Pool Layer:

This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast, reduces memory and also prevents from overfitting.

5. Fully-Connected Layer:

This layer is a regular neural network layer which takes input from the previous layer and computes the class scores and outputs the 1-D array of size equal to the number of classes.

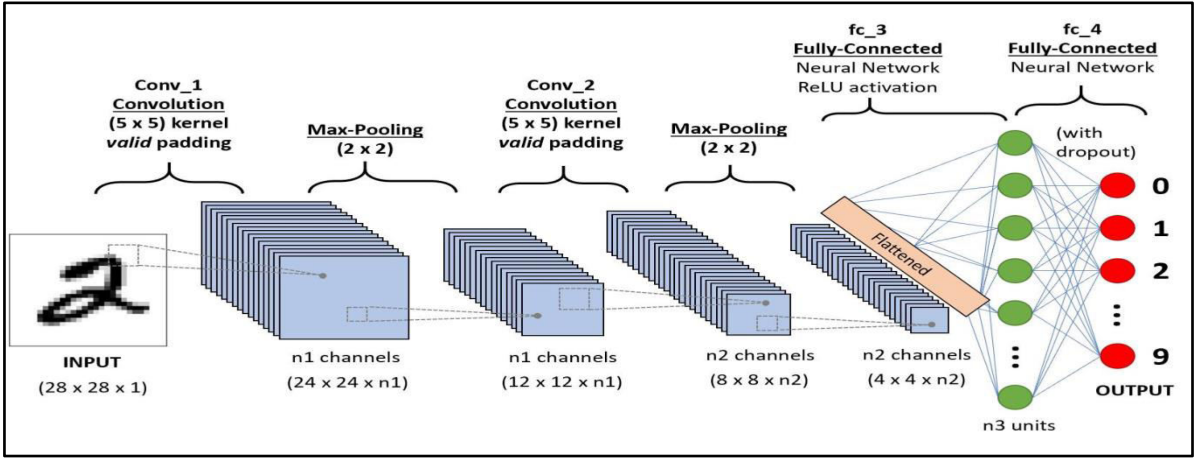

Fig.2: Basic CNN architecture showing alternating convolution and pooling layers.

4. IMAGE AUGMENTATION

Image augmentation is necessary when the dataset provided is less. If you have a lot of parameters, you would need to show your machine learning model a proportional amount of examples, to get good performance. Also, the number of parameters you need is proportional to the complexity of the task your model has to perform. There are ways by which we can achieve more data from the current dataset for training purposes.

1.Scaling:

When using real data set scaling is necessary as sometimes the image can be tiny or large or sometimes an object covers an entire image therefore scaling is an important part.

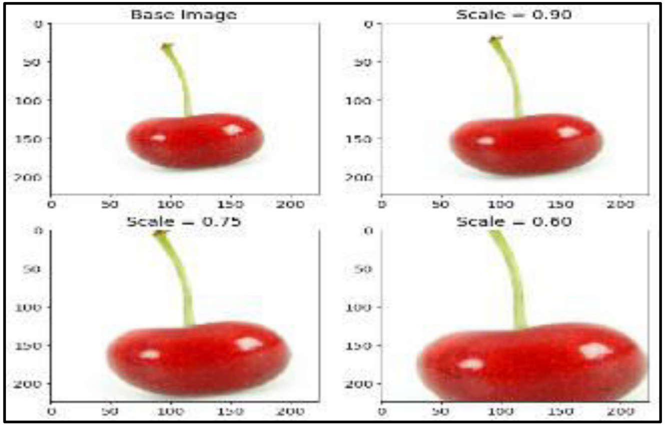

Fig.3: Image scaling.

2. Translation

We want our network to recognize the image in any part of the image, since the object can be partially present in corners or edges. For this reason we shift various parts of images.

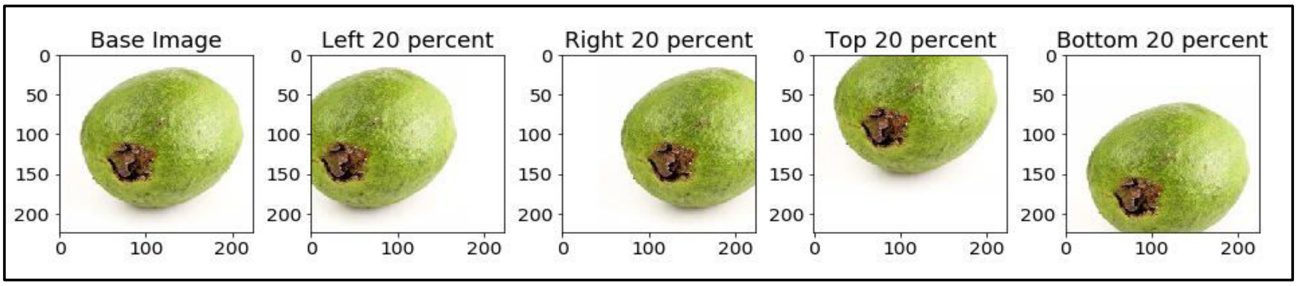

Fig.4: Image translation.

3. Rotation

This is done so that our network can recognize the image in any given orientation, that's why rotation is done.

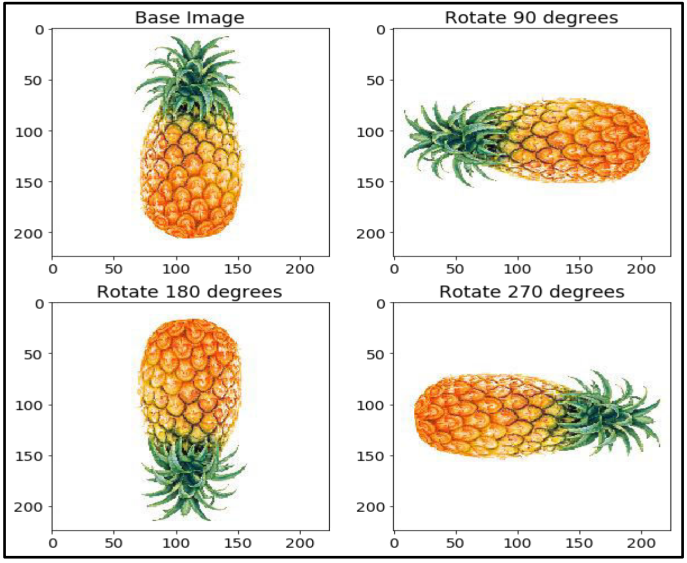

Fig.5: Image rotation.

5. LOSS FUNCTION

Loss function can be defined as the way to measure your algorithm or in other words to understand how well your algorithm works on your dataset. It tells you how good your prediction is on the dataset, if the value of the loss function is high we can say that predictions are bad and if the value is less we can say that predictions are good. In mathematical notation, it might look something like |(y_predicted - y)|.

For example if our y_redicted = 1000 and y = 100 our value of loss function will be high meaning that predictions are not good. There are many loss functions used depending on the problem statement.

1.Mean Squared Error(MSE):

Mean Squared Error (MSE) is a basic loss function: it's easy to understand and implement and generally it works pretty well on a number of models. To calculate MSE, we take the difference between your predictions and the truth, square it, and average it out across the whole dataset value. Mathematically it can be stated as below. Our goal is to minimize this mean function so that our model predicts better.

$$\text{LOSS} = \frac{1}{\square}\,\square(\square - \underline{\square})^2$$

2. Likelihood Loss:

Likelihood function is also a simple technique and is commonly used the problems based on classification. The function takes the predicted probability for each input example and multiplies them. And although the output isn't exactly human interpretable, it's useful for comparing models.

3. Cross entropy loss:

This function is also known as log loss and is used widely in classification problems. This function is a slight modification of the likelihood loss function by use of logarithm in this.
Below is the mathematical formula,

$$\text{LOSS} = -\frac{\square}{\square}\,\square(\square\ \square\square\square\ (\square) + (\square - \square)\ \square\square\square\ (\square - \square))$$

When the actual class is 1, the second half of the function disappears, and when the actual class is 0, the first half drops. That way, we just end up multiplying the log of the actual predicted probability for the ground truth class.

It has one more advantage that it fines penalty on the wrong prediction.
The graph below is when the true label =1, and you can see that it skyrockets as the predicted probability for label = 0 approaches 1. Therefore the penalty helps in optimization of your model.

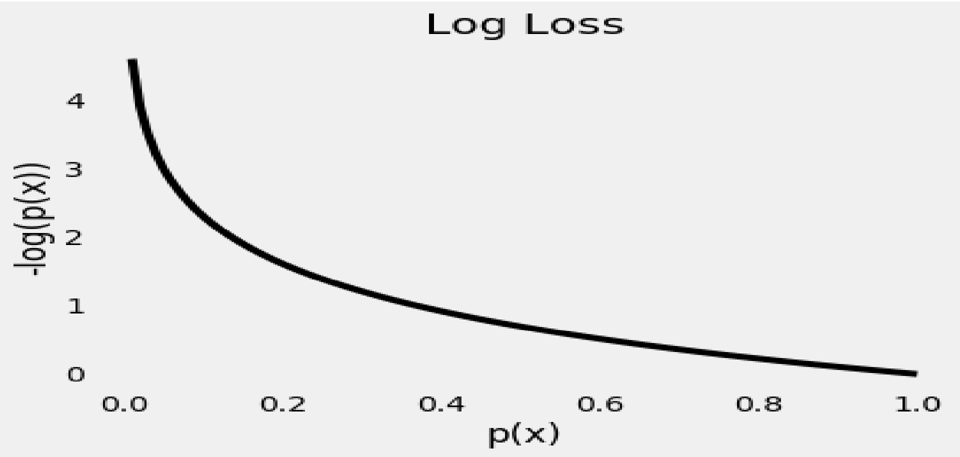

Fig.6: Log loss graph.

6. OPTIMIZATION TECHNIQUES

Optimization techniques help us minimize or maximize the error function which are basically the mathematical dependent function so that our model can learn. These leaned values are used to calculate the predicted value. Almost all the CNN networks have weight(W) and bias(b) these are the learnable parameters which are used in computing the output . These values are updated in order to minimize the loss. There are many different optimization algorithm present

1.Gradient Descent:

It is the most popular and widely used optimization algorithm used. There are basically two steps involved in this : forward propagation and backward propagation. In forward propagation we calculate the dot product of input to its weight with addition of bias and this output is then sent to the activation function like Relu , sigmoid , tanh, etc.

After this we propagate backwards in the Network carrying Error terms and updating Weights values using Gradient Descent, in which we calculate the gradient of Error(E) function with respect to the Weights (W) or the parameters , and update the parameters (here Weights) in the opposite direction of the Gradient of the Loss function w.r.t to the Model's parameters.

2. AdaGrad:

It simply allows the learning Rate to adapt based on the parameters. So it makes big updates for infrequent parameters and small updates for frequent parameters. For this reason, it is well-suited for dealing with sparse data. The best thing about this algorithm is that it uses a different value of learning rate for every parameter at any given time.

There are few disadvantages of this like its learning rate is always decreasing and it happens due to accumulation of each squared gradient, eventually it causes the learning rate to shrink and become so small. It becomes so small that the models stop learning and hence its learning speed decreases. This problem is solved by AdaDelta, an extension of AdaGrad which tends to remove the decaying learning Rate problem. Instead of accumulating all previous gradients, Adadelta limits the window of accumulated past gradients to some fixed size 'w'.

3. Adam:

Adam stands for Adaptive Moment Estimation. Adaptive Moment Estimation (Adam) is another method that computes adaptive learning rates for each parameter. It not only stores an exponentially decaying average of past squared gradients like AdaDelta but also keeps an exponentially decaying average of past gradients, just like momentum . We can consider Adam to be a mixture of momentum and AdaGrad.

7. DIFFERENT ARCHITECTURES OF CNN

1. **LeNet-5** (1998):

LeNet-5, a convolutional network produced by LeCun in 1998, that classifies digits, which is applied by many institutions to recognise hand-written numbers or images on greyscale input pictures.

It consists of five alternating layers of convolution and pooling, followed by two fully connected layers. LeNet-5 was used on a giant scale to mechanically classify hand-written digits.
The main limitation was a considerable procedure burden, specifically at that point of the time. Earlier GPUs and CPUs were not that developed to support numerous operations simultaneously.

2. **AlexNet** *(2012):*

AlexNet was the 1st deep CNN design that showed good results on tasks of classification of images and object recognition. AlexNet was developed by Krizhevesky, he increased the training capability of CNN by creating it more complex and by applying numerous parameter optimizations. In AlexNet, depth was extended from five (LeNet) to eight layers to create CNN applicable for various classes of pictures. It uses ReLu(Rectified Linear Unit) as the activation function, in place of Tanh or Sigmoid.

ReLu = f(x) = max(0,x)

The advantage of the ReLu over sigmoid is that it trains a large number of parameters quickly. The problem of vanishing weights is also solved by ReLu. Another advantage is that it resolved the matter of overfitting by adding a Dropout layer after each FC layer.
Some disadvantages are reshaping is required for general element-wise matrix multiplication and more memory requirement

3. **VGG-16** *(2014):*

VGG net may be a plain CNN design among all alternatives. Though it's straightforward, it does outdo several complicated architectures. It's the first challenger in the ImageNet Challenge in 2014. It scored 1st place on the image localization task and second place on the image classification task.VGG internet design was developed by Simonyan. The Visual pure mathematics cluster (VGG) fabricated the VGG-16 that has thirteen convolutional and three fully-connected layers, carrying with them the ReLU tradition from AlexNet.

Its advantages are that it showed sensible results each for image classification and localization issues and is convenient, effective and highly accurate. Some disadvantages are that this design isn't suitable for deeper networks, because of the Vanishing Gradients downside. The main limitation related to VGG was the employment of 138 million parameters, which made it tough to deploy it on low resource systems.

4. **Xception** *(2017):*

Xception is an architecture, which exploits the idea of depth wise separable convolutions. Exception was developed by François Chollet. Xception makes the network computationally economical by decoupling spatial and feature- map (channel) correlation.

Its features are that it makes computation simple by singly convolving every feature-map across spatial axes, that is followed by pointwise convolution (1x1 convolutions) to perform cross channel correlation, Xception has higher accuracy compared with most other models on the gradient descent steps.

5. **ResNeXt** *(2017):*

ResNeXt is an improvement over the Inception Network. A brand new term, cardinality, was introduced in this architecture. Cardinality is an extra dimension that refers to the dimensions of the set of transformations.

ResNeXt used the deep homogenized topology of VGG and simplified GoogleNet architecture by fixing spatial resolution to 3x3 filters among the split, transform, and merge Block.

8. DATASET INFORMATION

In the past few years, satellite companies' offerings to scientists have increased dramatically. Thousands of researchers now use high-resolution data from commercial satellites for their work. In fact, there are datasets that may overwhelm large scale computing clusters.

The National Aeronautics and Space Administration's Landsat is currently the leader amongst those providing datasets for research purposes.

For our research, we looked at a number of datasets: *USGS Earth Explorer, Sentinel Open Access Hub,Planet Labs,DigitalGlobe Open Data Program, Geo-Airbus Defense,National Institute for Space Research, Brazil, Satellite Land Cover*. While going through these datasets, we had to look at several parameters based on our research domain and our processing capabilities.

We needed the data to be in small chips, in order to train the model effectively. Another major requirement was the images should be already labeled since we have to train the model on thousands of images, which was manually not plausible for us. Also, if manual labeling is considered the question of authenticity arises, so for accurate results the best option was labeled images

The labels we chose most viable and generally occurring images in the Amazon rainforest and would be divided into three major categories:-

- Cloud cover labels: These include the different types of clouds and haze
- Common labels: These include water, habitation, agriculture, road, cultivation, etc
- Rare labels: These include mining, blooming, blowdown and other naturally occurring phenomena in the Amazon Rainforests

After a comparative study of the datasets provided by the different platforms, The datasets from Planet Labs, Geo-Airbus Defense and National Institute for Space Research, Brazil were the most suited for our study.

Datasets from USGS Earth Explorer and Digital Global platform were the largest in quantity and by far the sharpest but weren't labeled which meant we had to do it manually. Satellite Land Cover's datasets did not cover all the parameters required for this study.

Amongst the viable options, Geo-Airbus did not give the entire dataset for free and there were some proprietary conditions and National Institute for Space Research, Brazil gave us data only for the part of rainforest inside Brazil, which meant we had to discard these. Planet Labs data supplied us with 40,479 training images and their corresponding labels. These images are in GeoTiff format with four bands of images, i.e., red, blue, green and near-infrared images. Each image file is a 256x256 pixel "chip" which is sampled from a larger 6600x2200 pixel "Planetscope Scene".

9. IMPLEMENTATION DETAILS

Many frameworks are available for deep learning, of which Google's TensorFlow is the latest and fast growing. It is an open source software library for doing high computational jobs using data flow graphs where edges denote tensors. With this, a single API can be used to distribute load between multiple nodes (CPUs or GPUs). This library has been publicly available since November, 2015. Keras is considered to be the second fast growing deep learning framework. This is capable of running on top of TensorFlow or Theano. Theano is an open source Python library for numerical computations and simplifies the process of writing deep learning models. Another framework is Caffe, developed by the Berkeley Vision and Learning Center (BVLC). It has many worked examples of deep learning, written in Python. Giving more importance to GPUs is the framework called Torch, having an underlying C/CUDA implementation. Matlab's matconvnet and Torch's torch are also widely used frameworks for deep learning.

10. CONCLUSION

This paper focused on the effectiveness of convolutional neural networks for geographical and natural phenomena classification. It is useful in many applications like education, generating various government schemes for protection of these areas, and security by reducing illegal activities. This paper conveyed useful background knowledge to understand the problem domain along with different pre-processing techniques. It presented the CNN based approaches for solving this problem. In contrast to the traditional feature extraction based methods that are time consuming for selecting appropriate features for learning of models, CNN automatically learns those features efficiently and that is why CNN can become very convenient for real-world scenarios. We carried out an exhaustive survey and analysis of geographical and natural phenomena classification methods using CNN. Furthermore, we analysed CNN architectures of different works and suggested architecture as per the characteristics of the dataset.

11. ACKNOWLEDGEMENT

This paper and the survey behind it would not have been possible without the exceptional support of our guide, Professor Priya Surana. Her enthusiasm, knowledge and exacting attention to detail have been an inspiration and kept our work on track from our first encounter with the research papers of classification models to the final draft of this paper.

12. REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton,"ImageNet classification with deep convolutional neural networks", *Advances in Neural Information Processing Systems*, 2012.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] Karen Simonyan & Andrew Zisserman, “ Very Deep Convolutional Networks For Large-scale image recognition”, *ArXiv 1409.1556.,* 2014.

[5] Canziani, Alfredo and Paszke, Adam and Culurciello, “An Analysis of Deep Neural Network Models for Practical Applications”, *ArXiv* 1605.07678, 2017.

[6] Matthew D. Zeiler, Rob Fergus, “Visualizing and Understanding Convolutional Networks”, *13th European Conference, Zurich*, *Proceedings, Part I, Volume 8689, pages:818-833* ,2014.

[7] K. Jarrett, K. Kavukcuoglu, M. Ranzato and Y. LeCun, "What is the best multi-stage architecture for object recognition?," 2009 IEEE 12th International Conference on Computer Vision, Kyoto, pp. 2146-2153, 2009.

[8] Springenberg, Jost & Dosovitskiy, Alexey & Brox, Thomas & Riedmiller, Martin, “Striving for Simplicity: The All Convolutional Net”, *ICLR,* 2014.

[9] Glorot, X. and Bengio, Y., “Understanding the difficulty of training deep feedforward neural networks”, *Proceedings of AISTATS 2010*, volume 9, pp. 249-256, May 2010.

[10] LeCun, Y., Bottou, L., Orr, G., and Müller, K., “Efficient backprop”, *Neural networks: Tricks of the trade*, pp. 546-546, 1998.

[11] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9.

[12] Szegedy, Christian & Vanhoucke, Vincent & Ioffe, Sergey & Shlens, Jon & Wojna, ZB., “Rethinking the Inception Architecture for Computer Vision”, ArXiv:1512.00567 [cs.CV], 2016.

[13] L. Shao, F. Zhu and X. Li, "Transfer Learning for Visual Categorization: A Survey," in IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 5, pp. 1019-1034, May 2015.

[14] S. J. Pan, Q. Yang, "A survey on transfer learning", *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.

[15] W. Dai, Q. Yang, G.-R. Xue, Y. Yu, "Boosting for transfer learning", *Proc. 24th Int. Conf. Mach. Learn.*, pp. 193-200, Jun. 2007.

[16] Y. Zhang, Q. Hua, D. Xu, H. Li, Y. Bu and P. Zhao, "A Complex-Valued CNN for Different Activation Functions in Polarsar Image Classification," IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 2019, pp. 10023-10026.

[17] R. G. Mantovani, T. Horváth, R. Cerri, J. Vanschoren, A. C. de Carvalho, "Hyper-parameter tuning of a decision tree induction algorithm" in Intelligent Systems (BRACIS) 2016 5th Brazilian Conference on, IEEE, pp. 37-42, 2016.