INDIAN SIGN LANGUAGE RECOGNITION

Umesh Sati[1], Hitesh[2], and Shafali Dhall[3]

1 Bharati Vidyapeeth's College of Engineering, INDIA; umeshsati54@gmail.com.

2 Bharati Vidyapeeth's College of Engineering, INDIA; hk325690@gmail.com

3 Bharati Vidyapeeth's College of Engineering, INDIA; shafali.dhall@bharatividyapeeth.edu

* Correspondence: shafali.dhall@bharatividyapeeth.edu

Abstract:

The aim of this project is to recognize characters in Indian Sign Language(ISL). So much research has been done in the corresponding field of American Sign Language, but the same cannot be said for Indian Sign Language. Lack of standard datasets occluded features, and variation in the language with locality have been the major barriers that have led to little research being done in ISL. we have reported four-fold cross-validated results for the different approaches, and the difference from the previous work done can be attributed to the fact that in our four-fold cross-validation, the validation set corresponds to the images of a person contrasting from the persons in the training set.

Keywords: Convolutional Neural Network (CNN), Tensorflow, Sign language recognition, Keras, Inception v3 model, Mobilenet.

1. Introduction

This project aims at identifying alphabets in Indian Sign Language from the corresponding gestures as in [1]. Gesture recognition and sign language recognition has been a well researched topic for American Sign Language(ASL), but few research works have been published regarding Indian Sign Language(ISL). But instead of using high-end technology like gloves or Kinect, we aim to solve this problem using state of the art computer vision and machine learning algorithms. Communication is one of the basic requirements for survival in society. Deaf and dumb people communicate among themselves using sign language but normal people find it difficult to understand their language. Extensive work has been done on American sign language recognition but Indian sign language differs significantly from American sign language.ISL uses two hands for communicating(20 out of 26) whereas ASL uses a single hand for communicating. Using both hands often leads to the obscurity of features due to the overlapping of hands. In addition to this, lack of datasets along with variance in sign language with locality has resulted

in[2] restrained efforts in ISL gesture detection. Our project aims at taking the basic step in bridging the communication gap between normal people and deaf and dumb people using Indian sign language. Effective extension of this project to words and common expressions may not only make the deaf and dumb people communicate faster and easier with the outer world, but also provide a boost in developing autonomous systems for understanding and aiding them.

The Indian Sign Language lags behind its American Counterpart as the research in this field is hampered in [3] by the lack of standard datasets. Unlike American Sign Language, it uses both hands for making gestures which lead to occlusion of features. ISL is also subject to variance in the locality and the existence of multiple signs for the same character in [1]. Also, some characters share the same alphabet(E.g V and 2 have the same sign, similarly W and 3 have the same sign) and the resolution of the sign is context-dependent.

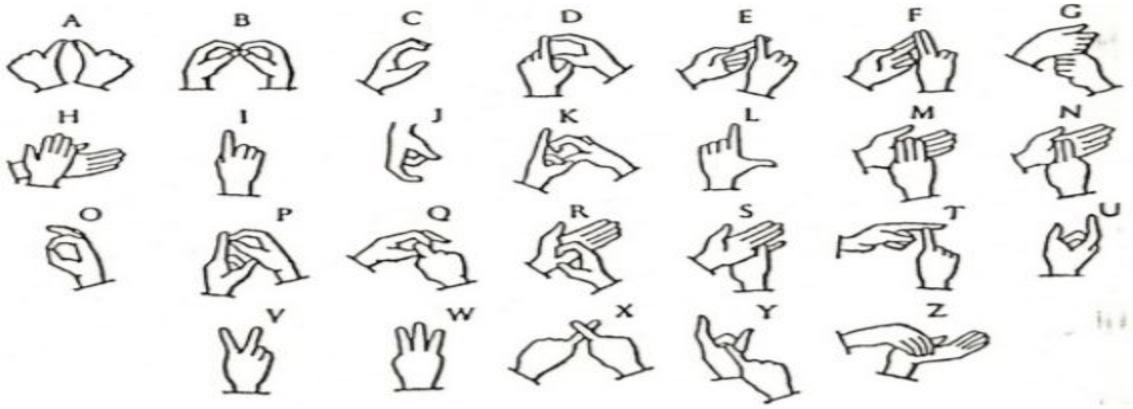

Figure 1: Indian Sign Language Gestures[1]

2. Literature Survey

2.1 Machine Learning

Little work has been done previously on ISL. One of the approaches included key point detection of Image using SIFT[2] and then matching the key point of a new image with the key points of standard images per alphabet in a database to classify the new image with the label of one with the closest match. Another one in[2] is calculated the eighteen vectors of the covariance matrix calculated from the vector representation of the image and used Euclidean distance of new image eigenvector with those in training data set to classify the new image. Some of them used Neural networks for training but their dataset comprised of only single-handed images and their feature vectors are based on the angle between fingers, a number of fingers, etc. The major problem with all these works that there was no mention of where the dataset was collected from and from the images it appeared as if it was taken from the webcam by the people working themselves. In a

thesis for related work, we found at, the authors had formed the dataset on one of their[4] team members and divided that into training and testing sets before reporting the accuracy. Also, they had tested only on a subset of the English alphabets.

2.1.1 CNN Architecture

In order to assess the strengths and limitations of CNNs, several architectures were trained and tested with a particular focus on a 22 layers deep model called GoogLeNe]t[3]. This very efficient network achieves state-of-the-art accuracy using a mixture of low-dimensional embeddings and heterogeneous sized spatial filters21. Increased convolution layers and improved utilization of internal network computing resources allow the network to learn deeper features. For example, the first layer might learn edges while the deepest layer learns to interpret hard exudate, a classification feature. The network contains convolution blocks with activation on the top layer that defines complex functional mappings between inputs and response variables, followed by batch normalization[7] after each convolution layer. As the number of feature maps increases, one batch normalization per block is introduced in succession.

The max-pooling sample-based discretization process was performed with kernel size 3x3 and stride 21. The network was then flattened to one dimension after the final convolutional block. The dropout of network layers was performed until reaching the dense five node output layer, which uses a softmax activation function to compute the probability of classification labels. Leaky rectified linear unit activation was also applied with gradient value 0.01 to mitigate dead neuron bottlenecks during back-propagation2. The network uses convolutional layer L2 regularization to reduce model overfitting, cross-entropy computed error loss and the Xavier method[3] of initializing weights so that neuron activation functions start out in unsaturated regions.

3. Preferred Models

3.1 Transfer Learning

Transfer Learning is an advanced technique of Deep Learning where a model developed for a task is used as a starting point for a model on a second context similar task. Training a Convolutional Neural Network(CNN) on ImageNet [give refer] dataset takes around 2-3 weeks across multiple GPUs. In the current research, two pre-trained models, Inception V3 model [5] and MobileNets open-source models for comparison purposes. Both the mentioned models utilize CNN at their cores as discussed further.

In the Transfer Learning technique, the last layers or some of the last layers of the pre-trained models, depending on the data being used are unfrozen and retrained on the new dataset to obtain

the required results. If only the last layer in the model is unfrozen, the resulting partial model then gives the probabilities of the outputs against each other rather than giving the most probable output, this altering will be much beneficial for sentence construction which will be discussed in the future work. If multiple last layers are unfrozen then a smaller Learning Rate is expected during retraining with the new dataset as this will help preserve the weights of the last few layers.

The MobileNets are small, low-latency, low-power models parametrized to meet the resources constraints of a mobile device. Google Research has 16 pre-trained open-source MobileNet models with varying parameters and accuracy competing for accuracy against time. MobileNets are mobile first Deepwise Separable Convolution Neural <. Networks, counting depthwise and pointwise convolutions as different layers, MobileNets has 28 layers and is trained on the ImageNet dataset

The Inception V3 model that is used in the current research is a successor of Inception V1 model which is a variant of GoogleNet[7]with the additional features of Batch Normalization and the introduction of Factorization of larger convolutions into smaller convolutions. The main theory of models with Inception architecture is to find out how an optimal local sparse structure in a convolutional vision network can be approximated and covered by readily available dense components.

3.2 MobileNet_v1_0.50_224 model

The MobileNet_v1_0.50_224 model has 1.24 million parameters. The top layer of mobilenet_v1_0.50_224 takes as input a 1001-dimensional vector for each image, we train a softmax layer on top of this representation. Assume that the softmax layer contains N labels, this corresponds to learning N+1001*N model parameters corresponding to the learned biases and weights. When the process of Transfer Learning is initiated on[6] the mentioned model, an analysis of all images is done and bottlenecks for each are calculated and stored, the bottleneck is referred to as the layer just before the final output layer that is responsible for the classification. This reasoning about why the training of the last layer of[2] alone still results in such a high accuracy is that the weights that are stored corresponding to each layer after forward and backward propagations during the long training durations are responsible for the classification of all the 1,000 objects in ImageNet will serve the same purpose with the new classes that we are aiming to classify with our provided data.

Our method as generalized showcases two variants of accuracy, Training accuracy, which is the measure of the percentage of images that were used in the current training batch and were correctly-labelled, another is Validation accuracy, which is the measure of precision on a randomly selected group of images from a varying set. It can be distinctly observed from the graphs in Fig. (4) and Fig. (6) that the validation accuracy fluctuates at[5] irregular intervals, this

is due to the reasoning that a random subset of the validation set is chosen for each validation accuracy measurement. Similar to as in Deep Learning training, the hyper-parameters are the ultimate game changers in Transfer Learning, several iterations were performed with varying hyperparameters, which in the present case includes, the Learning Rate (LR), which controls the magnitude of the update to the final layer during training.

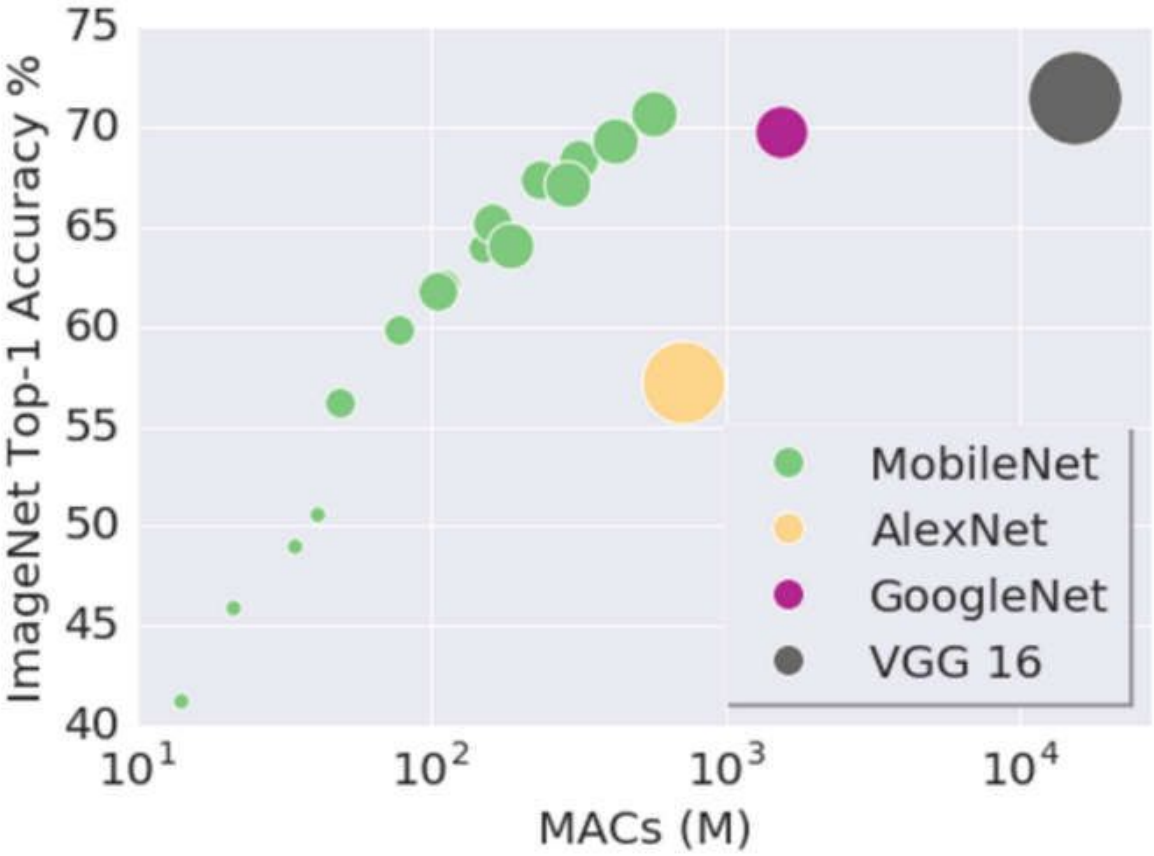

Figure 2: Comparisons of different models[4]**.**

3.3 Inception V3

There are 4 versions. The first GoogLeNet must be the Inception-v1 [4], but there are numerous typos in Inception-v3 [1] which lead to wrong descriptions about Inception versions. These maybe due to the intense ILSVRC competition at that moment. Consequently, there are many reviews in the internet mixing up between v2 and v3. Some of the reviews even think that v2 and v3 are the same with only some minor different settings.

Nevertheless, in Inception-v4 [5], Google has a much more clear description about the version issue:

*"The Inception deep convolutional architecture was introduced as **GoogLeNet** in (Szegedy et al. 2015a), here named **Inception-v1**. Later the Inception architecture was refined in various ways, first by the introduction of **batch normalization** (Ioffe and Szegedy 2015) (**Inception-v2**). Later by additional **factorization** ideas in the third iteration (Szegedy et al. 2015b) which will be referred to as **Inception-v3** in this report."*

Inception V3 by Google is the 3rd version in a series of Deep Learning Convolutional Architectures. Inception V3 was trained using a dataset of 1,000 classes from the original ImageNet dataset which was trained with over 1 million training images, the Tensorflow version has 1,001 classes which is due to an [7] additional "background' class not used in the original ImageNet. Inception V3 was trained for the ImageNet Large Visual Recognition Challenge where it was a first runner up.

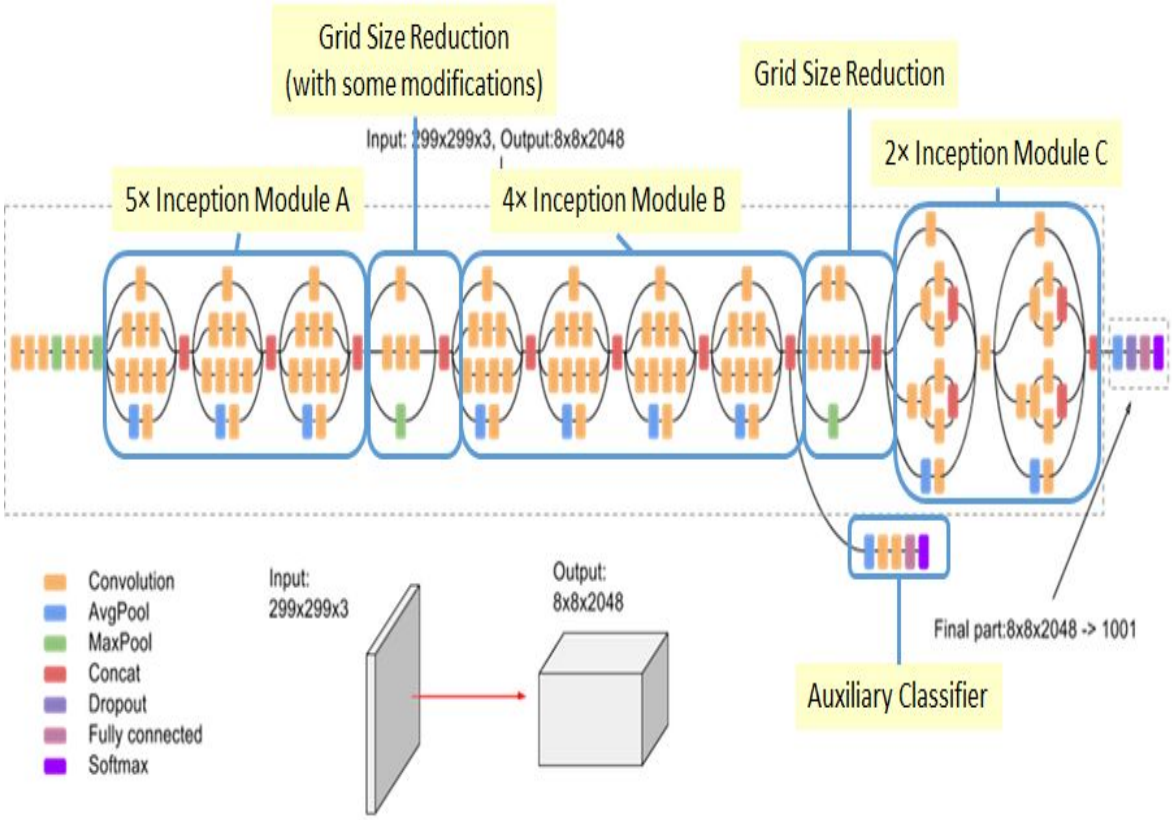

Figure 3: Inception V3 Architecture[5]

4. Results

Upto this point the whole discussion revolves around experimentation with MobileNet models, this is done with the aim to find the best model that will suit our needs, for the clarification of the same, we need to test the resulting models in real time environment and need a detailed

visualization of the accuracy results of both, Fig. (4) is the Accuracy graph of the Training set and the Validation set of the retrained Inception V3 model against training steps (50,000) and Fig. (5) represents the cross-entropy value of the retrained MobileNet model during training of the model, against training steps, colour coding of orange is inductive of Training data and Sky-blue that of Validation data, the final validation accuracy is 88.33%. The training accuracy, in this case, is 90% which indicates that the results are authentic and there is no instance of over-training.

Table 1: MobileNet Accuracy

S. No.	Model Name	Parameters (in million)	Time to retrain	Accuracy
1.	MobileNet	1.24	4 Hrs.	88.33 %

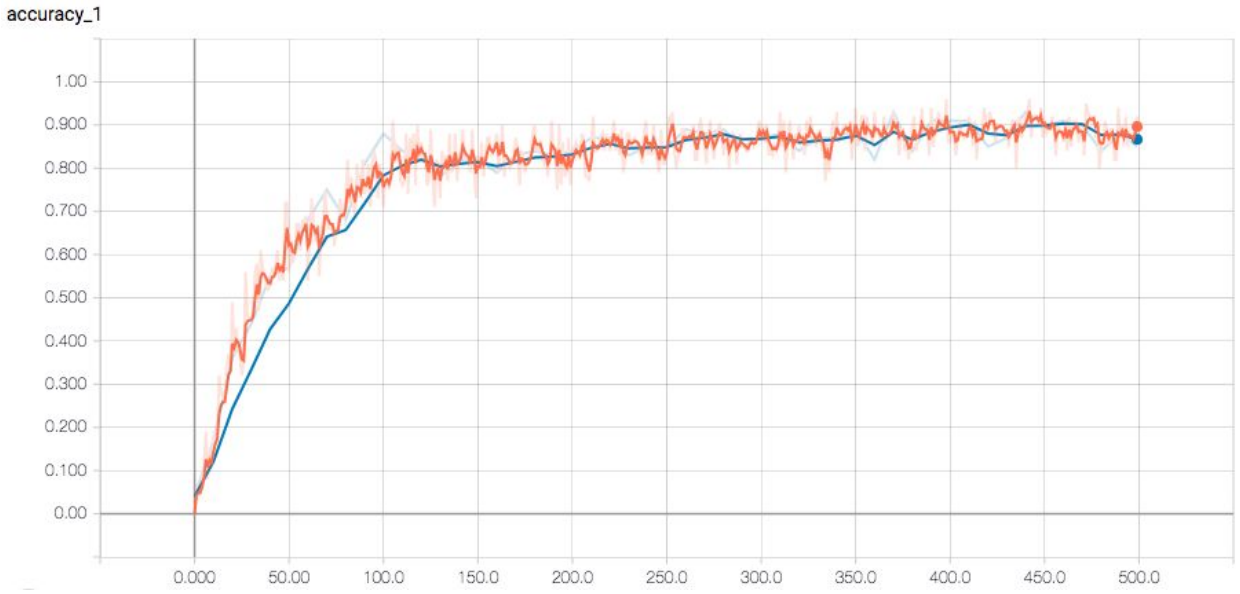

Figure 4: Accuracy Graph

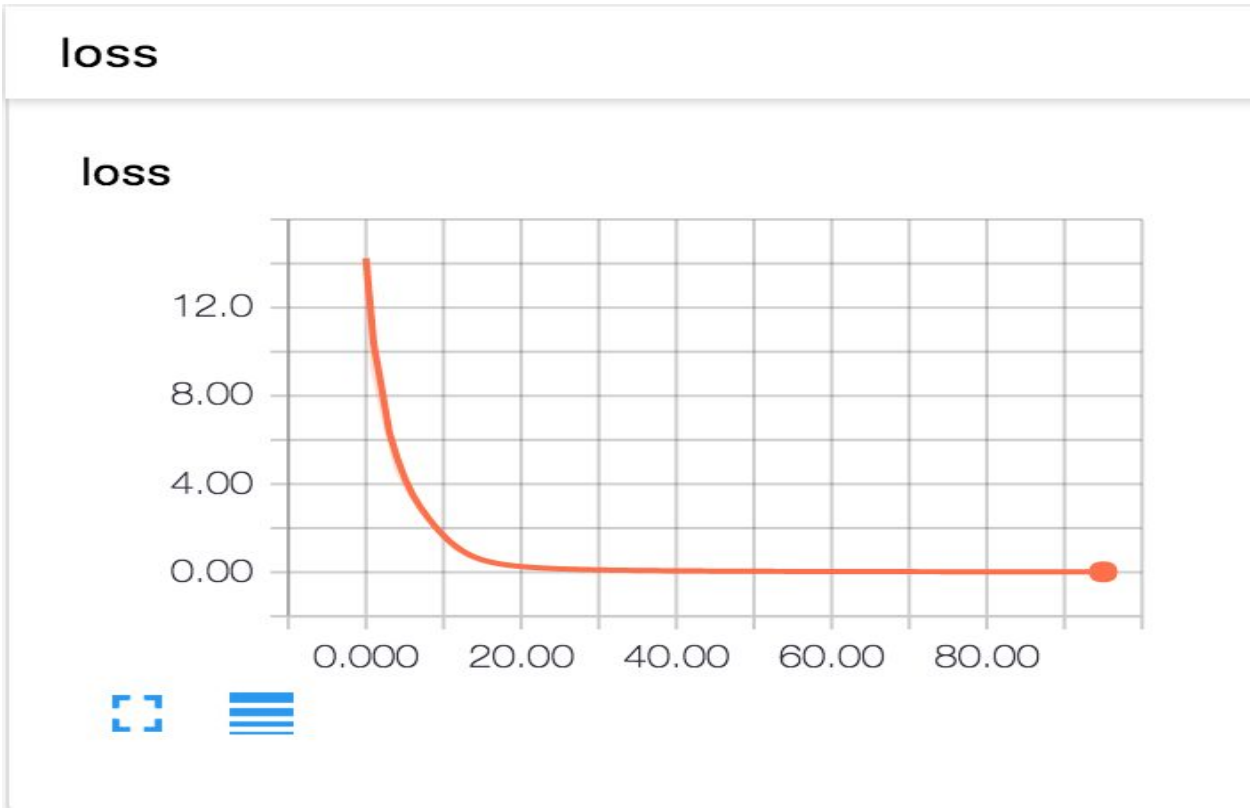

Figure 5: Loss Graph of MobileNet

References

[1] Indian Sign Language Character Recognition https://www.cse.iitk.ac.in/users/cs365/2015/_submissions/vinsam/report.pdf.

[2] Rathi, Dhruv. (2018). Optimization of Transfer Learning for Sign Language Recognition Targeting Mobile Platform.

[3] Lam C, Yi D, Guo M, Lindsey T. Automated Detection of Diabetic Retinopathy using Deep Learning. *AMIA Jt Summits Transl Sci Proc*. 2018;2017:147-155. Published 2018 May 18.

[4] Inception-v3 — 1st Runner Up (Image Classification) in ILSVRC 2015. https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c

[5] Shivashish Naithani, Shivam Bharadwaj, Dhiraj Kumar(2019). Automated Detection of Diabetic Retinopathy using Deep Learning.

[6] Batch Normalization (Inception-v2 / BN-Inception) —The 2nd to Surpass Human-Level Performance in ILSVRC 2015 (Image Classification). https://medium.com/@sh.tsang/review-batch-normalization-inception-v2-bn-inception-the-2nd-to-surpass-human-level-18e2d0f56651

[7] Ajadi, Olaniyi & Meyer, Franz & Webley, Peter. (2016). Change Detection in Synthetic Aperture Radar Images Using a Multiscale-Driven Approach. Remote Sensing. 8. 482. 10.3390/rs8060482.

[8] Jin, Xin & Chi, Jingying & Peng, Siwei & Tian, Yulu & Ye, Chaochen & Li, Xiaodong. (2016). Deep Image Aesthetics Classification using Inception Modules and Fine-tuning Connected Layer.

[9] Pratt, Harry & Coenen, Frans & Broadbent, Deborah & Harding, Simon & Zheng, Yalin. (2016). Convolutional Neural Networks for Diabetic Retinopathy. Procedia Computer Science. 90. 200-205. 10.1016/j.procs.2016.07.014.

[10] Detecting Invasive Ductal Carcinoma with Convolutional Neural Networks. https://software.intel.com/content/www/us/en/develop/articles/detecting-invasive-ductal-carcinoma-with-convolutional-neural-networks.html

[11] Inception V3 Deep Convolutional Architecture For Classifying Acute Myeloid/Lymphoblastic Leukemia. https://software.intel.com/content/www/us/en/develop/articles/inception-v3-deep-convolutional-architecture-for-classifying-acute-myeloidlymphoblastic.html