IMPROVING MODEL PERFORMANCE AND REMOVING THE CLASS IMBALANCE PROBLEM USING AUGMENTATION

A Project Report submitted in partial fulfilment of the requirements for the award of the degree of

MASTER OF TECHNOLOGY

IN
DATA SCIENCE

Submitted by

Allena Venkata Sai Abhishek, 122021601009

Under the esteemed guidance of

Dr. Venkateswara Rao Gurrala
Professor, CSE Dept

DEPARTMENT OFCOMPUTER SCIENCE & ENGINEERING
GITAM
(Deemed to be University)
VISAKHAPATNAM

June 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

DECLARATION

I, hereby declare that the project report entitled "**IMPROVING MODEL PERFORMANCE AND REMOVING THE CLASS IMBALANCE PROBLEM USING AUGMENTATION**" is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of M.Tech. in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date: 09-06-2022

Registration No	**Name**	**Signature**
122021601009	Allena Venkata Sai Abhishek	

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

CERTIFICATE

This is to certify that the project report entitled "**IMPROVING MODEL PERFORMANCE AND REMOVING THE CLASS IMBALANCE PROBLEM USING AUGMENTATION**" is a bonafide record of work carried out by **Allena Venkata Sai Abhishek (122021601009)** students submitted in partial fulfillment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

Project Guide	**Head of the Department**
Dr. Venkateswara Rao Gurrala	**Dr. R. Sireesha**
Professor	**Professor**

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

I am are very much thankful to my guide **Dr. Venkateswara Rao Gurrala**, Professor in Department of Computer Science & Engineering, who has been very concerned and has aided me with all the materials essential for the project work and preparation of the project report.

I express my deep sense of gratitude and thanks to **Dr. R Sireesha**, Professor and Head of the Department of CSE for her guidance and for expressing her valuable and graceful opinions in the project for its development and also for providing the lab sessions and extra hours to complete the project.

I would like to take this opportunity to express our profound sense of gratitude to the coordinator **Dr. John Jaidhan B**, Associate Professor and **Dr. Laxman Sahoo,** Senior Professor, Department of Computer Science and Information Technology for their constant motivation and support in allowing me to utilize the college resources there by facilitating the successful completion of our thesis.

I also take this opportunity to express my heartfelt credit to the teaching and non-teaching staff of the department, for their perspective comments and suggestions.

CONTENTS

FIGURES

TABLES

ABSTRACT

Artificial intelligence can create fantastic outcomes in the lab, yet data scientists struggle to reproduce these results on real-world data. And it’s no surprise – real-world data reflects the messy world that created it, containing gaps and biases. The information in reality comprises of different sorts of difficult highlights. A significantly found one is the class irregularity wherein the quantity of models in various classes in a dataset is inconsistent. It is being resolved using various sampling techniques on the data, out of which data augmentation techniques and random sampling are used for oversampling the data of minority classes. This paper aims to improve the model performance while removing the class imbalance problem using various augmented balanced datasets. They are generated using a custom-built balanced dataset generator based upon the AugStatic library. AugStatic is a custom-built image augmentation library with lower computation costs than other image augmentation libraries. The augmented balanced datasets are inputted to the RESNET18 model. The best test accuracies are found for various class imbalance ratios for three classification datasets - Intel Scene, CIFAR10, and Chest X-Ray Pneumonia dataset, for 100 epochs, stored and analyzed. The research analysis has brought up some astounding insights into the model performance for various augmentation techniques.

Keywords: Image classification, Deep Learning, ResNet18, Augmentation, Class Imbalance

1. INTRODUCTION

1.1 MOTIVATION

There are many augmentations supported by various augmentation libraries that help in improvement of the accuracies of the various models. To find the best augmentation technique based on accuracy using a model by building an augmentation library/ package/module, which will generate an augmented balanced dataset, in-turn, solving the Class imbalance problem.

1.2 PROBLEM DEFINITION

Analyzing various augmentation techniques used for improving the model performance while removing the class imbalance problem using various augmented balanced datasets, that are generated using a custom-built balanced dataset generator based upon the Custom-built Augmentation library – AugStatic.

1.3 OBJECTIVE OF THE PROJECT

Following are the objectives –
To find the best Data augmentation technique for three datasets based on the accuracies for the augmented datasets with balancing of unbalanced dataset using custom-built balanced dataset generator based upon the Custom-built Augmentation library – AugStatic.

1.4 APPLICATIONS

This system is utilized for tackling the problems, which arise in our day-to-day life in artificial intelligence and computer vision. Some of the scopes are:

1. This efficient system can be utilized to balance an unbalanced dataset using various augmentation techniques and to enhance the model’s accuracy.
2. It applies to any data set, which has an imbalance in the number of examples in the classes.
3. This can be used in various driverless cars technology are else any kind of automation which includes artificial intelligence.
4. It can be used to train the deep learning model much more efficiently with a small amount of data.
5. This saves us time and resources for finding more data
6. The class distribution’s imbalance ratio may vary. The severe imbalance

challenges the model greater and there can be specialized techniques requirement.

7. This problem statement is prominent in real-world problems related to data classification problems that poses an distribution in imbalance method,which are spam detection, fraud detection and churn prediction.

2. LITERATURE SURVEY

2.1. Data collection

The public databases to evaluate the algorithms are:

2.1.1. Intel Scenery Dataset

This dataset contains the picture information of scenes of nature all over the world. This information consists about twenty five thousand images of size with six classes of size 150x150. The classes are forest, buildings, mountain, glacier, street and sea.

2.1.2. CIFAR10

The dataset named CIFAR-10 contains sixty thousand thirty-two cross thirty-two color that has ten classes of images, with six thousand pictures per class. There are ten thousand test pictures and fifty thousand preparation pictures.

2.1.3. chest_X_ray_pneumonia

The dataset is coordinated into three envelopes and has two classes – Pneumonia and Normal. Fifty eight hundred images in X-Ray format. There is first evaluation for quality control by eliminating all inferior quality or indiscernible outputs. Two master doctors then reviewed the determinations for the pictures prior to being cleared for preparing the AI framework. To represent any reviewing mistakes, the assessment set was likewise really looked at by a third master.

2.2. Dataset Preparation

2.2.1 Intel Scenery Dataset

The proposed framework was prepared and tried utilizing datasets specifically Intel Scene Dataset. This is picture information of Scene of nature all over planet. This Data contains around twenty five thousand pictures of one fifty cross one fifty size less than various classes that are six. For our trial, we have involved two classes with the accompanying number of models in each class

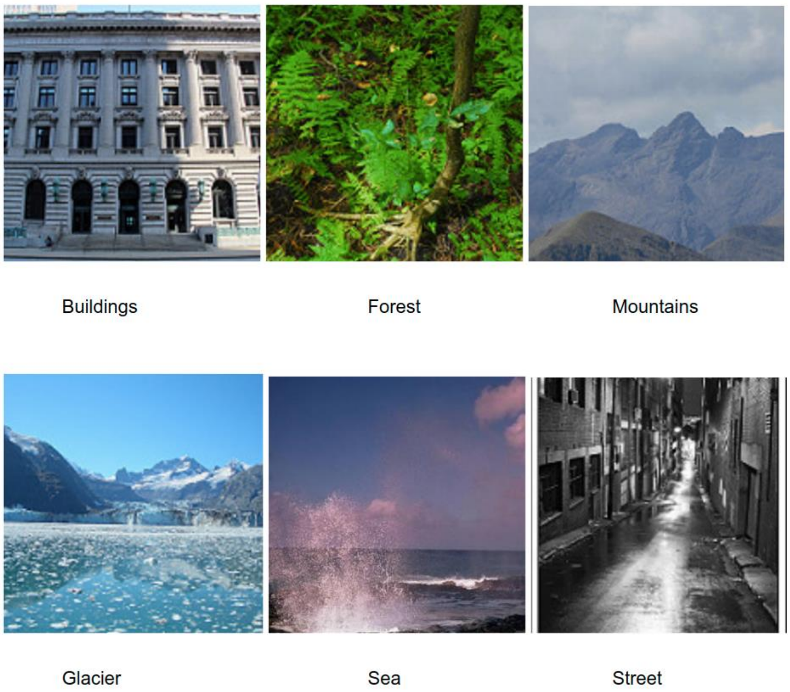


Figure. 1 Intel Scene Dataset

Training sets

A. BALANCED (1:1)

a) A = 2000 ; B = 2000

B. SLIGHT IMBALANCE (<=1:10) -

a) A = 2000 ; B = 1000

b) A = 2000 ; B = 500

c) A = 2000 ; B = 250

C. SEVERE IMBALANCE (>1:10) -

a) A = 2000 ; B = 125

b) A = 2000 ; B = 100

CLASS A = buildings

CLASS B = sea

The test set has 432 images in each of the two classes - buildings & sea. We standardized the appearances to 72 pixels. In view of the construction of pictures of 150*150 pixels unique pictures were resized to 224*224. Subsequent to resizing the pictures to 224*224, We are changing it into tensor and normalizing them with mean with x with 0.485, y with 0.456 and z with 0.406 and standard deviation, std with x with 0.229, with y with 0.224, with z with 0.225 separately. We are stacking the information into the information loader and going them through the model, which is a leftover organization RESNET18.

2.2.2 CIFAR10

The proposed framework was prepared and tried utilizing datasets in particular Intel Scene Dataset. The dataset named CIFAR-10 contains sixty thousand thirty-two cross thirty-two color that has ten classes of images, with six thousand pictures per class. There are ten thousand test pictures and fifty thousand preparation pictures. For our examination, we have involved 2 classes with the accompanying number of models in each class

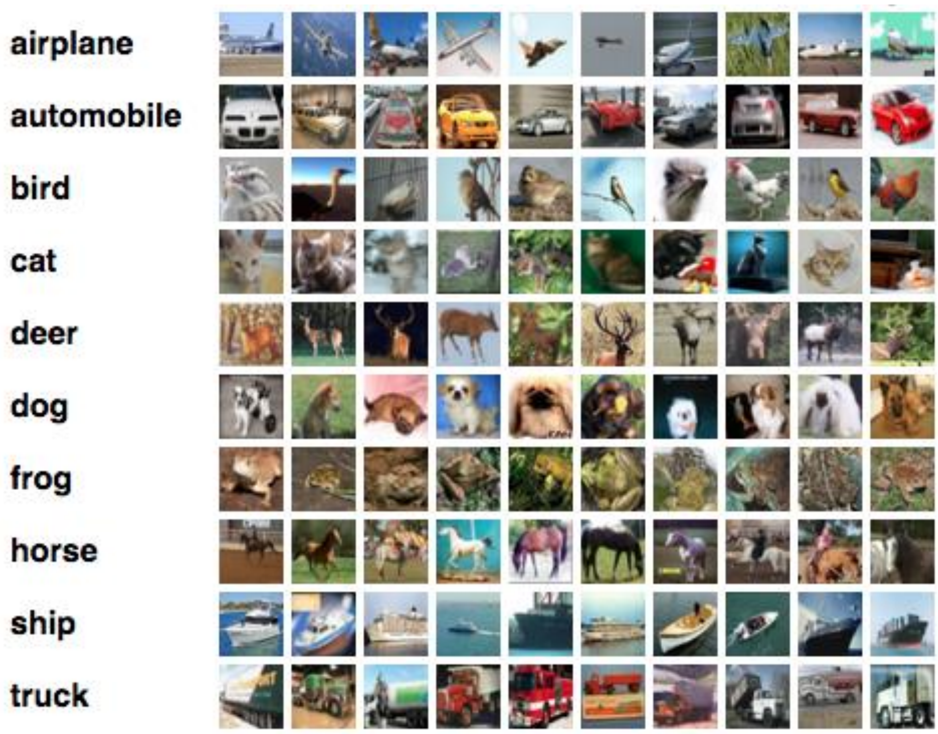


Figure. 2 CIFAR 10 Dataset

Training sets

A. BALANCED (1:1)

a) A = 5000 ; B = 5000

B. SLIGHT IMBALANCE (<=1:10) -

a) A = 5000; B = 2500
b) A = 5000; B = 1250
c) A = 5000; B = 625

C. SEVERE IMBALANCE (>1:10) -

a) A = 5000; B = 312
b) A = 5000; B = 250

CLASS A = dog
CLASS B = cat

The test set has 1000 images in each of the 2 classes - dog & cat. We standardized the countenances to 72 pixels. In light of the design of pictures of 150*150 pixels unique pictures were resized to 224*224. In the wake of resizing the pictures to 224*224, we are changing over the lem into tensor and We are changing it into tensor and normalizing them with mean with x with 0.485, y with 0.456 and z with 0.406 and standard deviation, std with x with 0.229, with y with 0.224, with z with 0.225 separately. We are stacking the information into the information loader and going them through the model, which is a lingering network RESNET18.

2.2.3. chest_X_ray_pneumonia

The dataset is coordinated into three envelopes and has two classes – Pneumonia and Normal. Fifty eight hundred images in X-Ray format. There is first evaluation for quality control by eliminating all inferior quality or indiscernible outputs. Two master doctors then reviewed the determinations for the pictures prior to being cleared for preparing the AI framework. To represent any reviewing mistakes, the assessment set was likewise really looked at by a third master. For our examination, we have involved two classes with the accompanying number of models in each class

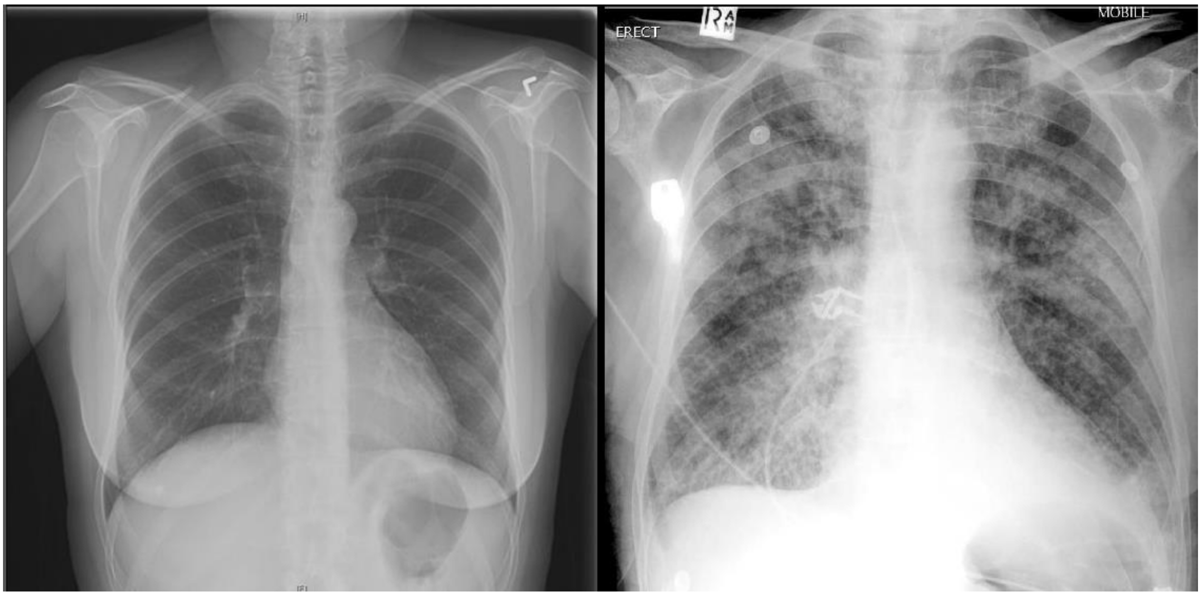


Figure. 3 Chest X-Ray Pneumonia

A. BALANCED (1:1)

a) A = 1200; B = 1200

B. SLIGHT IMBALANCE (<=1:10) -

a) A = 1200; B = 600

b) A = 1200; B = 300

c) A = 1200; B = 150

C. SEVERE IMBALANCE (>1:10) -

a) A = 1200; B = 75

b) A = 1200; B = 60

CLASS A = NORMAL

CLASS B = PNEUMONIA

The test set has 1000 pictures in every one of the 2 classes - NORMAL and PNEUMONIA. We standardized the countenances to 72 pixels. In view of the design of pictures of 150*150 pixels unique pictures were resized to 224*224. In the wake of resizing the pictures to 224*224, we are changing over the lem into tensor and we are changing it into tensor and normalizing them with mean with x with 0.485, y with 0.456 and z with 0.406 and standard deviation, std with x with 0.229, with y with 0.224, with z with 0.225 separately

Database	Sample Details
Intel Scene Dataset	Sample taken - { 'buildings' -> 0, 'sea' -> 1 }
CIFAR10	Sample taken - { ‘dog’ -> 0, ‘cat’ -> 1 }

Chest X Ray_Pneumonia	Test taken - { 'normal' -> 0, 'pneumonia'-> 1 }

Table 1: Datasets

2.3. Imbalance & Classification of Unbalanced datasets

AI is a part of man-made brainpower, which is worried about the development and investigation of frameworks that can gain from information. There are three methodologies in the AI model for example administered learning, semi-regulated learning and unaided learning. For instance, a profound learning model could be prepared on pictures to figure out how to recognize person on foot and non-walker pictures. Subsequent to learning, it can then be utilized to order new pictures into person on foot and non-walker envelopes.

Key Terminologies -

Classification - Classification prescient displaying includes anticipating a class mark for an offered viewpoint.

Imbalanced Classification - A characterization prescient where there isn't equivalent classes.

Majority and minority Class - The class or the classes in an imbalanced arrangement prescient displaying issue that has numerous models or that has not many models.

Imbalanced classification problem - An grouping that is imbalanced is an issue is an illustration of an arrangement issue where the conveyance of models across the realized classes is one-sided or slanted. The conveyance can shift from a slight predisposition to an extreme irregularity where there is one model in the minority class for hundreds, thousands, or millions of models in the larger part class or classes.

Binary and Multiclass Classification Problem - A characterization prescient demonstrating issue where all models have a place with single of two classes and all models have a place with one of three classes.

Training Dataset - various models gathered from the issue area that incorporate the info perceptions and result class names. The issue and the kinds of models we might decide to utilize.

Causes of Class Imbalance - This could include real data being clumsy and inclinations presented during information assortment, and blunders made during information assortment.

- Biased Sampling.
- Measurement Errors.

Slight and Severe Imbalance - An imbalanced characterization issue in which the dissemination of models is lopsided just barely in the preparation dataset (for example 4:6) and the circulation of models is lopsided overwhelmingly in the preparation dataset (for example 1:100 or more).

Overfitting - Good execution on the preparation information, unfortunate speculation to different information. An assortment that is the canny examining techniques have been created trying to adjust these compromises.

Underfitting - Poor execution on the preparation information and unfortunate speculation to different information.

2.4. Traditional techniques for tackling class imbalanced data

Tending to class lopsidedness with conventional AI strategies has been concentrated on broadly throughout recent many years. The predisposition towards the larger part class can be eased by changing the preparation information to diminish awkwardness. Thusly, strategies for taking care of class unevenness are gathered into information level procedures, calculation level techniques, and cross breed draws near.

Data level techniques – It is for tending to class imbalance that is incorporated by

- **random over sampling**
- **random under sampling**

These procedures adjust the preparation circulations to diminish the degree of irregularity or lessen commotion, for example mislabeled examples or irregularities.

Advance under sampling techniques - It means to safeguard important data for learning.

- **Near-Miss algorithms -** There are a few Near-Miss calculations that utilization a K-closest neighbors (K-NN) classifier to choose larger part tests for expulsion in light of their separation.
- **One-sided selection –** It eliminates improperly classified tests of the preparation part, to eliminate greater part tests from class limits.

Over-sampling techniques - various informed over-inspecting procedures have additionally been created to fortify class limits, diminish over-fitting, and further develop segregation.

- **Synthetic Minority Over Sampling Method - SMOTE**
- **Borderline-SMOTE**
- **Safe-Level-SMOTE**

Extensive Experiments - He contrasted various seven testing methods and 11 normally utilized AI calculations. Each model was assessed with 35 benchmark informational collections utilizing six unique execution measurements to think about.

Algorithmic level techniques - Unlike information inspecting techniques, for dealing with class irregularity don't adjust the preparation information circulation. All things being equal, the learning or choice cycle is changed such that builds the significance of the positive class. Most regularly, calculations are changed to think about a class punishment or weight or the choice edge is moved in a manner that decreases predisposition for the other class.

- **Cost sensitive learning**

Table 2 Cost matrix

From: Survey on deep learning with class imbalance

	Actual positive	Actual negative
Predicted positive	$C(1,1) = c_{11}$	$C(1,0) = c_{10}$
Predicted negative	$C(0,1) = c_{01}$	$C(0,0) = c_{00}$

Figure.4 cost matrix

Ling and Sheng sort cost-touchy strategies as either an immediate strategy.

2.5. Data Augmentation Techniques for tackling Class Imbalance:

Data augmentation is a procedure to make new preparation information from existing preparation information misleadingly. Information expansion is a procedure, which builds a diversity of data available for preparing models, without really gathering new info.

Figure.5 Data Augmentation Techniques

2.6. Background research about Deep Learning and RESNET18

Computer vision - Computer vision is an interdisciplinary logical field that arrangements with how PCs can be made to acquire undeniable level comprehension from advanced pictures or recordings. According to the point of view of designing, it looks to computerize errands that the human visual framework can do. Deep has empowered the field of Computer Vision to progress quickly over the most recent couple of years. Here, I might want to talk about one explicit errand in Computer Vision called Semantic Segmentation. Despite the fact that specialists have thought of various ways of tackling this issue, I will discuss a specific engineering in particular ResNet, which utilize a Fully Convolutional Network Model for the errand. Beginning from a coarse grained down to an all the more fine grained understanding, we should portray these issues beneath:

a. Image classification

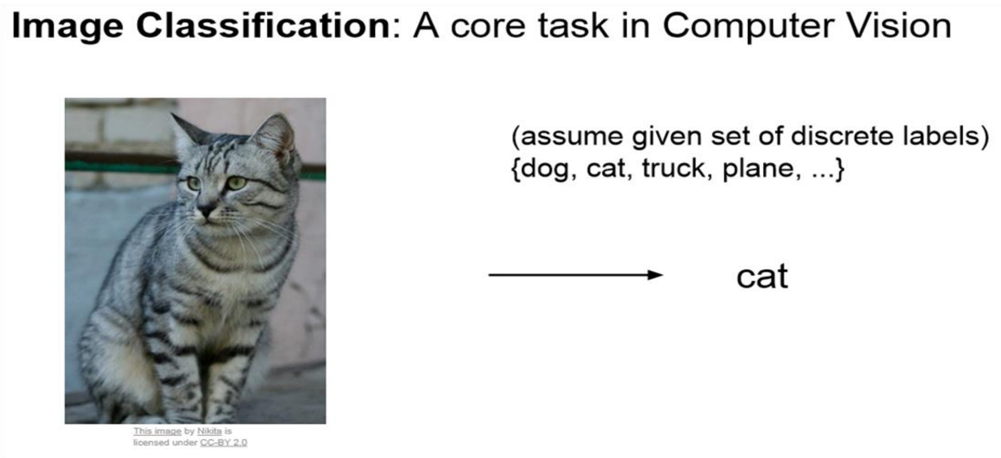


Figure.6 Image Classification

b. Classification with Localization

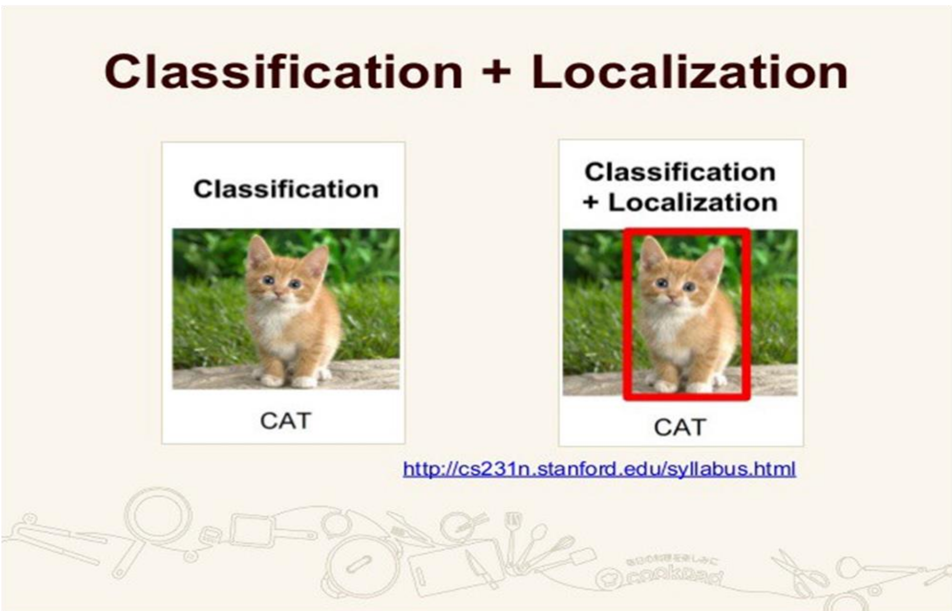


Figure.7 Classification and Localization

In restriction alongside the discrete name, we additionally expect the register to limit where the very object is available in the picture. This limitation is normally carried out utilizing a jumping box which can be recognized by a mathematical boundaries regarding the picture limit. Indeed, even for this situation, the supposition that is to have just a single item for every picture.

c. Object Detection

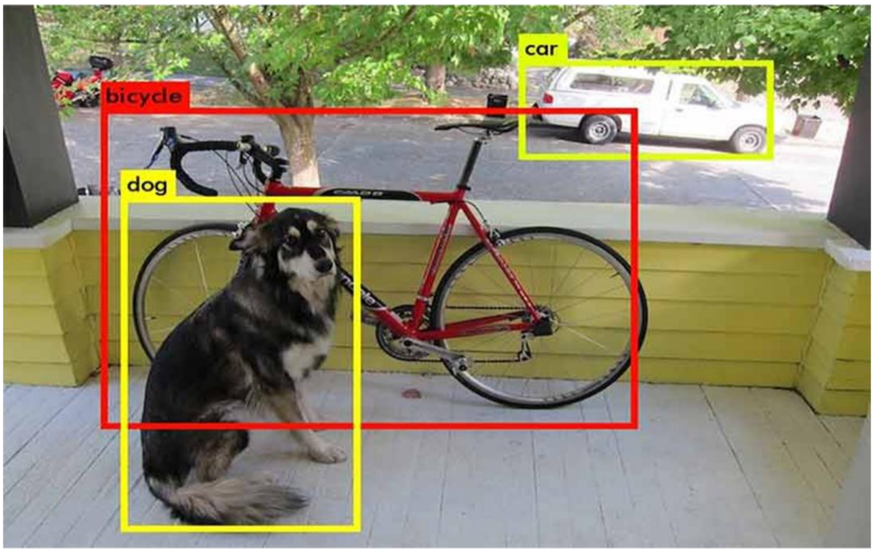


Figure.8 Object Detection

Object Detection stretches out restriction to a higher level where presently the picture isn't compelled to have just a single item, however can contain numerous items. The undertaking is to group and restrict every one of the articles in the picture. Here again the restriction is finished utilizing the idea of jumping box.

d. Semantic Segmentation

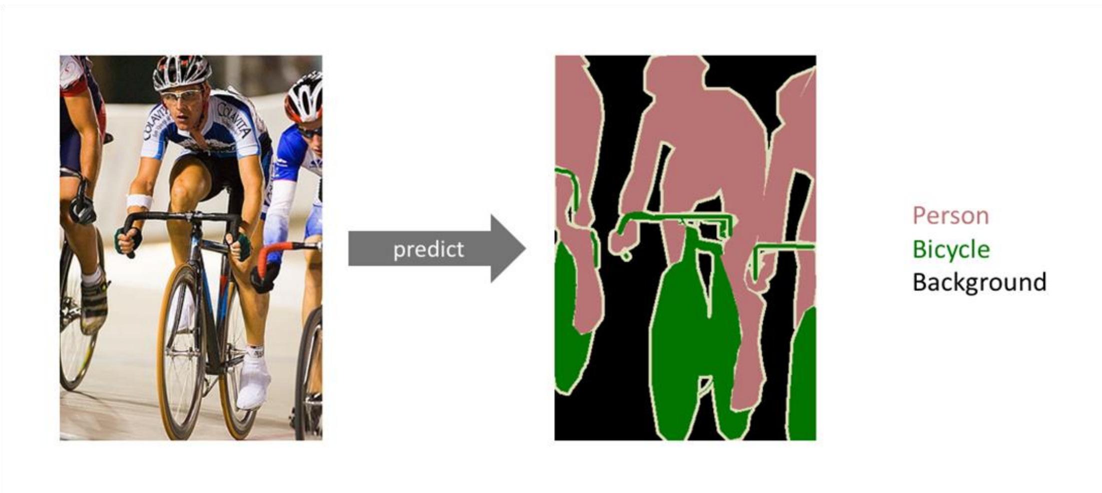


Figure.9 Semantic Segmentation

e. Instance segmentation

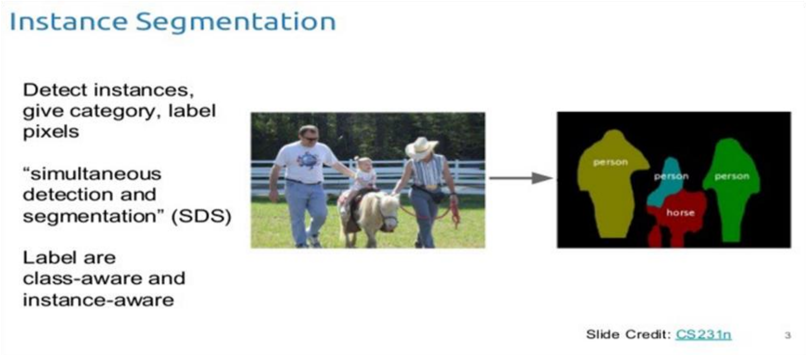


Figure.10 Instance Segmentation

Yet befuddled between the distinctions of item location, semantic division and occurrence division, underneath picture will assist with explaining the point:

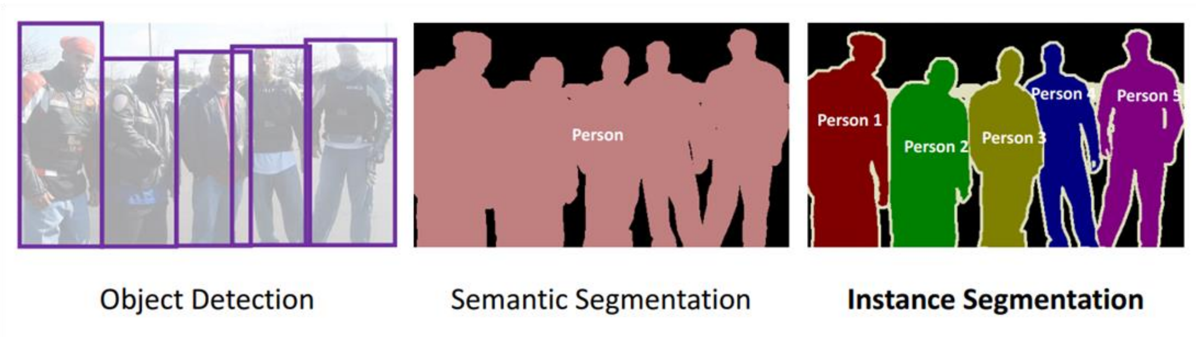


figure.11 Difference between object detection, semantic and instance segmentation

Structure of the Neural Network - The organization we have discussed so far should have a design or model that would help all the learning and computational cycles. A model that has various layers of neurons or hubs. MLP brain network advanced by back spread calculation depends on regulated system, for example the organization develops a model in view of instances of information with known yield. As like any organization models this organization additionally contains two fundamental layers called input layer and result layer.

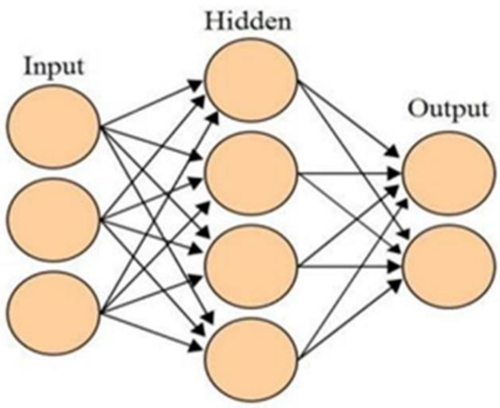


Figure.12 Artificial Neural Network with input, hidden and output layers

The result layer might contain at least one neurons relying upon the engineer and the framework. In the middle of between these two layers there might be at least one extra layers called the secret layers. We have added one secret layer in our framework since it increments exactness while the educational experience continues. There is no decent rule for the quantity of neurons that would be in the secret layer. These lots of layers are interconnected and as a feature of MLP the data streams unidirectional from input layer to yield layer through secret layer. In every hub there are alloted weight contingent upon the hub type. A glorification of a NEURON processes exercises or signals.

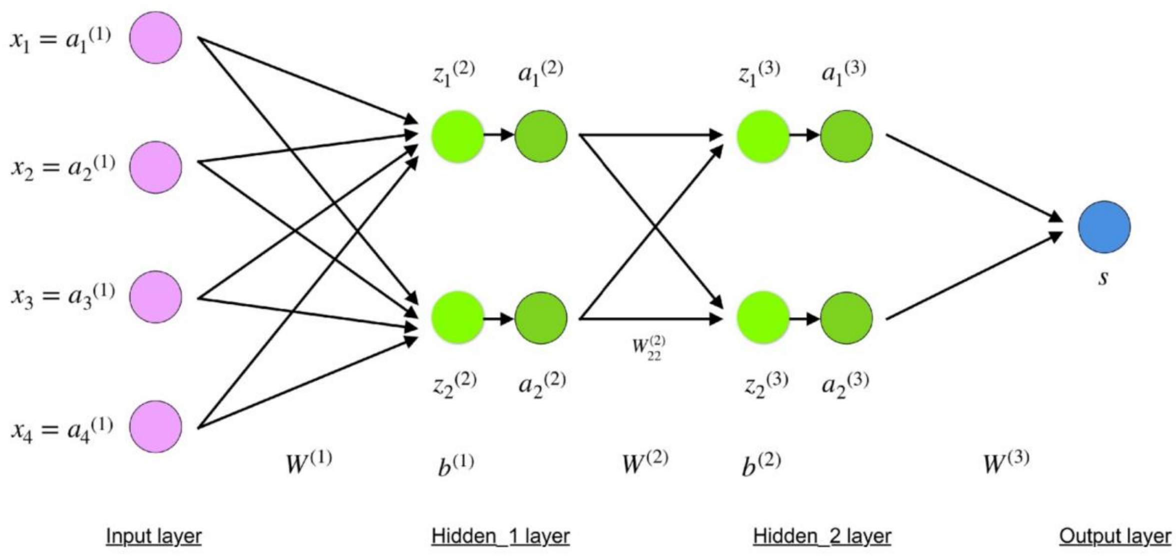
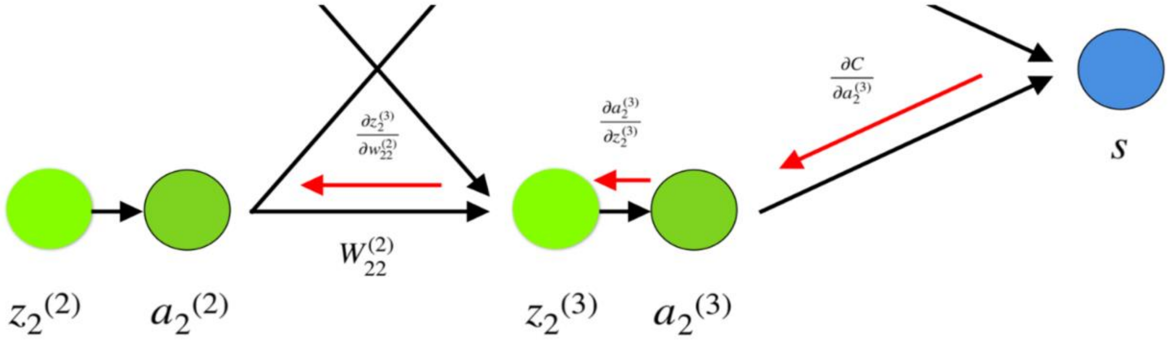


Figure.13 Hidden Layers



Weight (w_22)² connects (a_2)² and (z_2)², so computing the gradient requires applying the chain rule through (z_2)³ and (a_2)³:

$$\frac{\partial C}{\partial w_{22}^{(2)}} = \frac{\partial C}{\partial z_2^{(3)}} \cdot \frac{\partial z_2^{(3)}}{\partial w_{22}^{(2)}} = \frac{\partial C}{\partial a_2^{(3)}} \cdot \frac{\partial a_2^{(3)}}{\partial z_2^{(3)}} \, . \, a_2^{(2)} = \frac{\partial C}{\partial a_2^{(3)}} \, . \, f'(z_2^{(3)}) \, . \, a_2^{(2)}$$

Figure.14 Chain Rule

Neural Network - It is a strong and effective approach to getting an answer utilizing design acknowledgment. It is a broadly utilized strategy to perceive designs where in my exploration I have utilized it to decide feeling by recollecting the example of the lips locale.

Artificial Neural Network (ANN) - Biological human brain neural structure is more like this organization. They are integral assets for displaying, particularly when the basic information relationship is obscure. ANN can recognize and learn comparative examples between input informational collections and relating objective qualities. Subsequent to preparing ANN can be utilized to foresee the result of new autonomous information. As we referenced ANN follows the growing experience of the human cerebrum and can handle issues including nonlinear and complex information regardless of whether the information are loose and boisterous. It comprises of basic computational units called neurons, which are profoundly interconnected.

Convolutional Neural Networks - Convolutional Neural Networks .Sounds like an unusual blend of science and math with just the right amount.

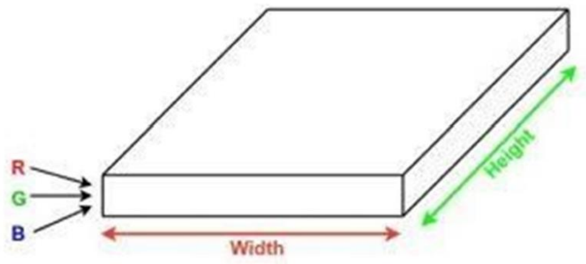


Figure.15 Dimension of image

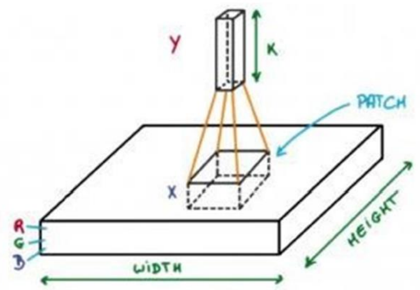


Figure.16 convolution process

Tensors: In straightforward words, it's simply a n-layered cluster in PyTorch. Tensors support a few extra improvements, which make them special: Apart from CPU, they can be stacked or the GPU for quicker calculations. On setting .requires_grad = True they begin shaping a retrogressive chart that tracks each activity applied on them to work out the inclinations utilizing something many refer to as a unique calculation diagram.

A straightforward DCG for duplication of two tensors would seem to be this:

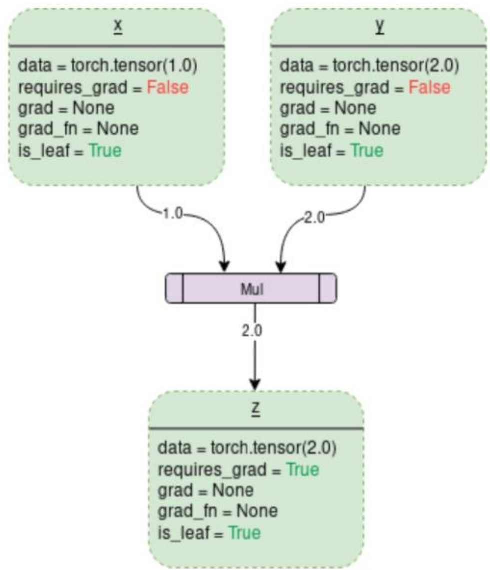


figure. 17 DCG Multiplication for two numbers

Loss Function - Loss function measure how far an expected worth is from its actual worth.

Activation Function -
f [Sum{(weight * input)} + bias]

Bias - Bias is very much like a catch included a straight condition. It is an extra boundary in the Neural Network which is utilized to change the result alongside the weighted amount of the contributions to the neuron. In addition, the predisposition esteem permits you to move the actuation capacity to one or the other right or left.

Forward Pass - The "forward pass" alludes to the computation cycle, upsides of the result layers from the information sources information. It's navigating through all neurons from the first to the last layer.

Backward Pass - then "in reverse pass" alludes to the most common way of including changes in loads (true picking up), utilizing slope plummet calculation (or comparable). Calculation is produced using the last layer, in reverse to the main layer.

Iteration - Backward and forward passes make together one "iteration".

Batch / mini-batch - During one iteration, you typically pass a subset of the informational collection, which is classified as "mini-batch" or "batch" (nonetheless, "batch" can also mean a whole set, thus the prefix "mini")

Epoch - Epoch implies passing the whole informational collection in groups. One age contains (number_of_items / batch_size) iterations.

Vanishing Gradient Problem -

- **Issue -** As additional layers utilizing specific enactment capacities are added to brain organizations, the inclinations of the misfortune work approaches zero, making the organization hard to prepare.
- **Why -** Certain initiation capacities, similar to the sigmoid capacity, crunches an enormous information space into a little info space somewhere in the range of 0 and 1. Hence, an enormous delta of contribution. Subordinate turns out to be little. Be that as it may, when n stowed away layers utilize an initiation like the sigmoid capacity, n little subordinates are increased together. Hence, the inclination diminishes dramatically as we proliferate down to

the underlying layers.

- **Step by step instructions to be aware in the event that the model has these issues -**
    - The model will improve gradually during the preparation stage and likewise conceivable preparation stops early, implying that any further preparation doesn't work on the model.
    - The loads nearer to the result layer of the model would observer to a greater extent a change though the layers that happen nearer to the info layer wouldn't change a lot (if by any means).
    - Model loads contract dramatically and become tiny while preparing the model.
    - The model loads become 0 in the preparation stage

Exploding Gradient Problem -

- **Problem -** In an organization of n stowed away layers, n subordinates will be duplicated together. If huge, the slope will increment dramatically as we engender down the model until they at last detonate, and this is the thing we call the issue of detonating angle.
- **Instructions to be aware assuming that the model has these issues -**
    - The model is not learning a lot of on the preparation information consequently bringing about an unfortunate misfortune.
    - The model will have huge changes in misfortune on each update because of the models unsteadiness.
    - The models misfortune will be NaN during preparation.
    - Model loads develop dramatically and become extremely huge while preparing the model.
    - The model loads become NaN in the preparation stage.
    - The derivatives are continually
- **Solution -**
    - ReLU - which doesn’t cause a little derivative.
    - Lessening how much of Layers
    - **Gradient Clipping -** If the angle gets excessively enormous, we rescale it to keep it little. All the more definitively, if **$\|g\| \geq c$**

      $c \cdot g/\|g\|$

      where c is a hyperparameter, g is the slope, and $\|g\|$ is the standard of g. Since $g/\|g\|$ is a unit vector, in the wake of rescaling the new g will have standard c. Note that in the event that $\|g\| < c$, then we don’t need to do anything.
    - **Residual networks -** as they give leftover associations directly to prior layers. The

leftover association straightforwardly adds the worth toward the start of the block, x, to the furthest limit of the block (F(x)+x). This lingering association doesn't go through enactment works that “squashes” the derivatives, bringing about a higher in general subsidiary of the block.

Degradation Problem -

If a “shallow” model can accomplish an exactness, their more profound partners ought to essentially have a similar precision. Be that as it may, when the model gets further, it turns out to be increasingly more challenging for the layers to proliferate the data from shallow layers and the data is lost.

RESNET18

For taking care of the problem of the evaporating inclination and Degradation issue, this design has been came upon to light with an idea named Residual Network. In this organization we utilize a procedure called skip associations. The skip association skips preparing from a couple of layers and interfaces straightforwardly to the result.

Residual Learning -

- Let g(x) be the capacity educated by the layers. How about we consider h(x) = g(x)+x, layers with skip associations.

Skip connection -

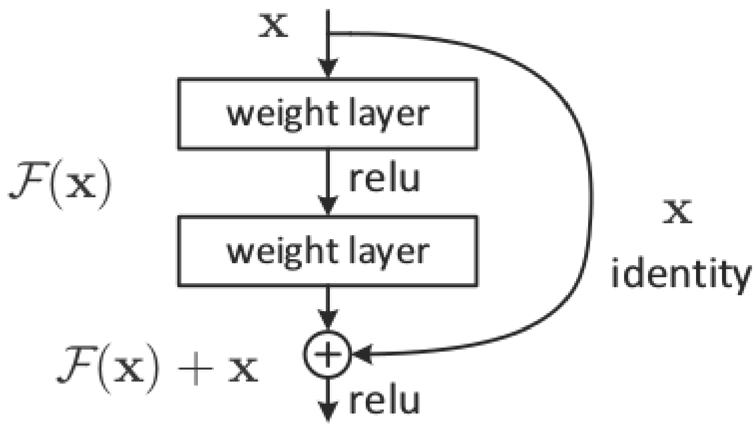


figure.18 Skip Connection

Identity Block (Input size = Output size) **-** Identity Block is utilized when there is no adjustment of information and result aspects. Convolutional block is practically equivalent to the personality block yet there is a convolutional layer in the easy route way to simply change the aspect with the end goal that the component of information and result matches. Here is identity block:

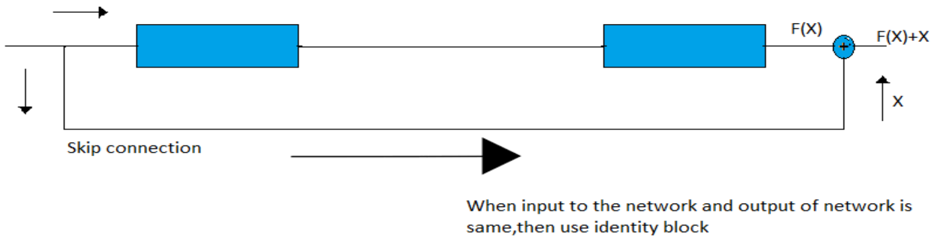


Figure.19 Identity Block

Convolutional Block (Input size != Output size) **-** It comprises of the Convolution Layer and the Pooling Layer. This layer shapes the fundamental part of Feature-Extraction. The Fully Connected Block — Consists of a completely associated straightforward brain network designxs .

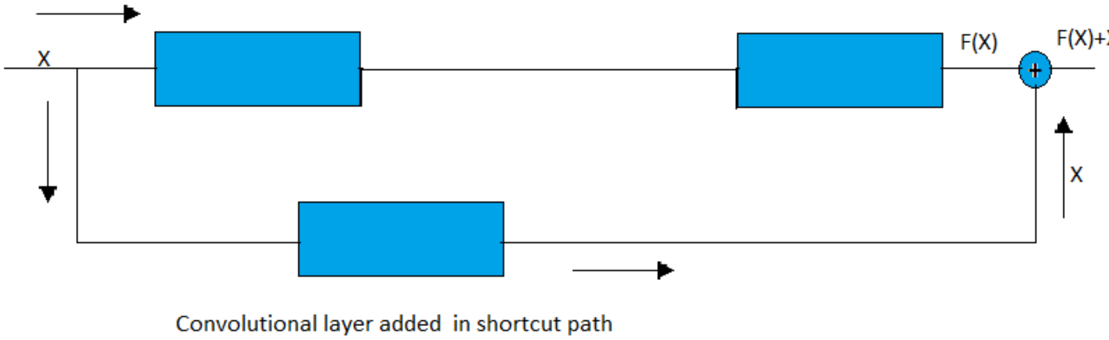


Figure.20 Convolutional Block

Convolutional Operation - We do component wise increase utilizing a bit on the information picture and it is utilized for include extraction.

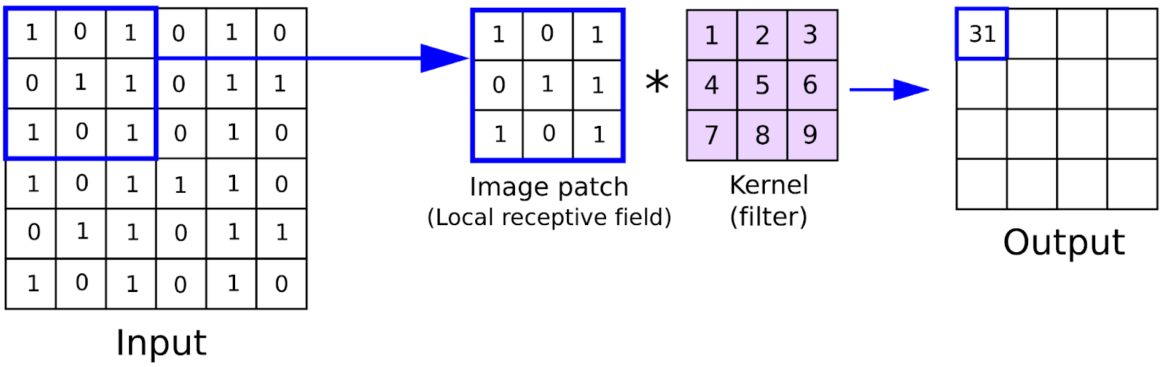


Figure.21 Convolutional Operation

Pooling Operation - Pooling lessens the size of the component map to assist us with involving the less boundaries in the organization i.e dimensionality decrease. Pooling makes the actuation portrayal (got from convolving the channel over the information mix of info and weight values) more modest and more reasonable.

3 types –

- Min - Pixel with Minimum value is selected
- Max – Pixel with Maximum value is selected
- Average – Average of the pixel value is taken

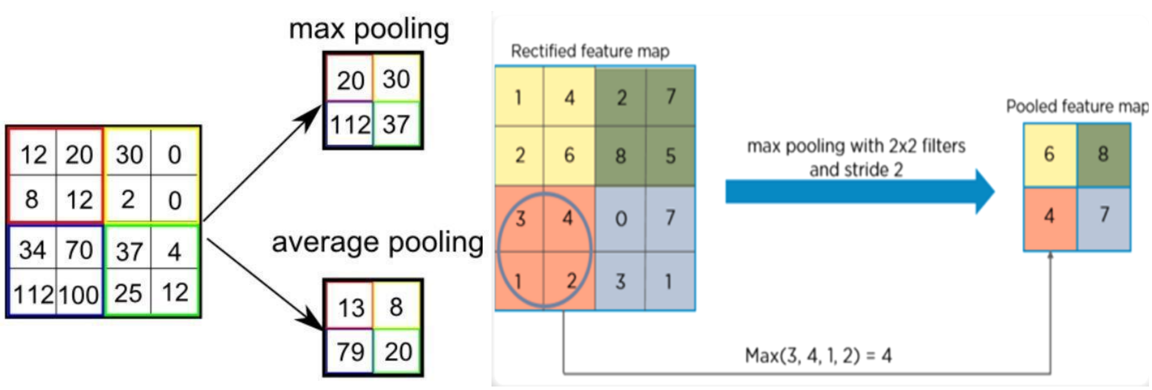


Figure.22 Pooling Operation

Creating and training a neural network involves the following essential steps:

- Characterize the engineering
- Forward engender on the design utilizing input information

- Compute the misfortune
- Backpropagate to compute the inclination for each weight
- Update the loads utilizing a learning rate

CUDA -Compute Unified Device Architecture - CUDA is an equal figuring stage created by NVIDIA and presented in 2006. It empowers programming projects to perform computations utilizing both the CPU and GPU.

Batch Normalization - Batch standardization is a method for making extremely complex neural networks, which normalizes the contributions to a particular layer for each and every small bunch.

Optimizers - Optimizers are calculations or techniques used to change the qualities of the brain organization like loads and learning rate to diminish the misfortunes.

Adam Optimizer - Adam is a substitution streamlining calculation for stochastic slope plunge for preparing profound learning models. Adam consolidates the best properties of the AdaGrad and RMSProp calculations to give an enhancement calculation that can deal with scanty inclinations on loud issues. Adam was joining the upsides of two different augmentations of stochastic inclination plunge.

Loss Function - Loss capacities characterize what a decent expectation is and isn't a method for estimating how well the model is performing.

NLL Loss Function - The negative log probability misfortune. Preparing an order issue with C classes is helpful. Whenever gave, the discretionary contention weight ought to be a one dimensional Tensor relegating weightage towards every class.

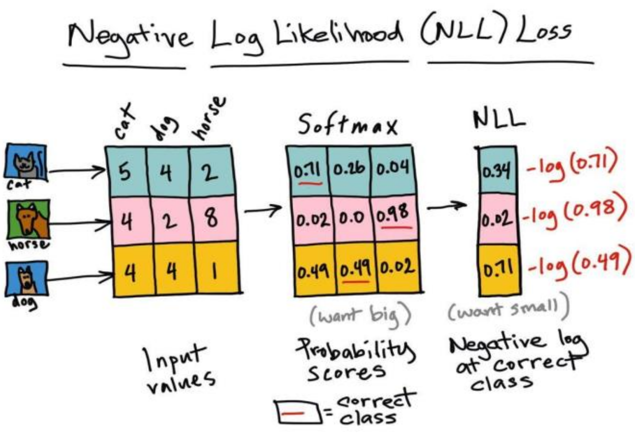


Figure.23 NLL Loss Function

2.7. Literature Review

Sl. No.	Journal Type and year	Authors	Title	Summary
1	IEEE, 2017	Mateusz Buda, Atsuto Maki, Maciej A. Mazurowski	A systematic study of the class imbalance problem in convolutional neural networks	As opposed to some classical machine learning models, oversampling does not cause overfitting of CNNs
2	IEEE, 2015	Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun	Deep Residual Learning for Image Recognition	Residual networks are easier to optimize, and can gain accuracy from considerably increased depth.
3	IEEE, 2019	Connor Taghi M. Khoshgoftaar	A survey on Image Data Augmentation for Deep Learning	Comparative study of various augmentation techniques
4	IEEE, 2020	Wanwan Zheng Mingzhe Jin	The Effects of Class Imbalance and Training Data Size on Classifier Learning	Naïve Bayes, logistic regression model is less susceptible to class imbalance while they have relatively poor predictive performance
5	IEEE, 2017	Marcus D. Bloice, Christof Stocker, Andreas Holzinger	Augmentor: An Image Augmentation Library for Machine Learning	Augmentor makes it easier to perform artificial data generation, by providing a stochastic, pipeline-based API.
6	IEEE, 2018	Alexandr A. Kalinin, Vladimir I. Iglovikov, Eugene Khvedchenya,	Albumentations: fast and flexible image augmentations	A fast and flexible library for image augmentations with many various image transform operations available, that is also an easy-to-use wrapper

		Alex Parinov, Alexander Buslaev		around other augmentation libraries
7	IEEE, 2018	Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, David Lopez-Paz	MixUp augmentation for image classification	Mixup trains a neural network on convex combinations of pairs of examples and their labels. Turn on screen reader support
8	IEEE, 2019	Zhiting Hu, Bowen Tan, Ruslan Salakhutdinov, Tom Mitchell, Eric P. Xing	Learning Data Manipulation for Augmentation and Weighting	The resulting algorithms significantly improve the image and text classification performance in low data regime and class-imbalance problems.
9	IEEE, 2021	Shanchuan Lin, Linjie Yang, Imran Saleemi, Soumyadip Sengupta	Robust High-Resolution Video Matting with Temporal Guidance	Input is down sampled for the encoder-decoder network, consists of an encoder that extracts individual frame's features, a recurrent decoder that aggregates temporal information, Deep Guided Filter module for high-resolution up sampling. Then DGF is used to up sample result.
10	Springer, 2021	Xu Sun, Huihui Fang, Yehui Yang, Dongwei Zhu, Lei Wang, Junwei Liu, Yanwu Xu	Robust Retinal Vessel Segmentation from a Data Augmentation Perspective	Data augmentation modules, namely, channel-wise random Gamma correction (correction on each channel) and channel-wise random vessel augmentation. (Morphological transformations on fine grained vessels.)

Table 2: Literature Review

3. TOOLS USED

3.1 Software Requirement Specification -

Requirement analysis is essentially classified into two kinds::

3.1.1. Functional requirements – The framework should respond to specific clients that act for specific circumstances.

3.1.2. Non Functional requirements - Nonfunctional prerequisites are necessities that are not straightforwardly worried about the predefined work conveyed by the framework. They might connect with developing framework properties, for example, dependability, reaction time and store inhabitance. A portion of the nonfunctional necessities related with this framework are thus beneath:

a. **Reliability -** Reliability in light of this framework effect of the system, right distinguishing proof of the looks and assessing pace for looking the acknowledgment of any information pictures.
b. **Ease of Use -** The framework is basic, easy to use, illustrations UI carried out so anybody can utilize this framework with practically no trouble.

3.2 Feasibility Study -

Prior to beginning the task, a plausibility study is finished for making a new framework that is cordial with the expense and other benefits with time and efforts. Following achievability-

3.2.1 Technical Feasibility - Technical attainability is quite possibly the earliest review that should be led after the task has been distinguished. Specialized achievability study incorporates the equipment and programming gadgets. The necessary advancements (Python language and Anaconda IDE) existed.

3.2.2 Operational Feasibility -

3.2.2.1 The augmentation is applied on the minority class to adjust the classes
3.2.2.2 The framework will order the images into classes.

3.2.3 Economic Feasibility - The motivation behind monetary. The system is financially attainable.

3.2.3.1. Intel Scene Dataset

3.2.3.2. CIFAR 10 Dataset

3.2.3.3. Chest X-Ray Pneumonia Dataset

3.2.4 Schedule Feasibility - The system viewed as possible in light of the fact that the framework is planned so that it will complete recommended time.

3.3. Hardware / Software Requirement -

3.3.1 Software Requirement -

a. **Python3.9 - programming language**
b. **Anaconda IDE / Google Colab (selective)**
c. **Windows OS**
d. **Libraries -**

i. **Numpy -** NumPy, which represents Numerical Python

ii. **PIL (Pillow Image Library) -** Python Imaging Library is a free and open-source controlling, and saving various picture record designs. The most recent adaptation of PIL is 1.1.

iii. **OpenCV -** OpenCV is written in C and C++ while PIL is composed utilizing Python and C, subsequently from this data, OpenCV appears to be quicker.

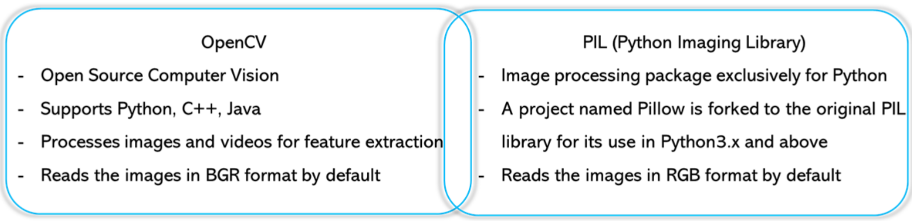


Figure.24 Difference b/w OpenCV and PIL

iv. **Tensorflow -** Currently, the most renowned profound learning library on the planet is Google's Tensor-Flow. TensorFlow is an open source AI system for all designers. It is utilized for carrying out AI and profound learning applications. To create and investigate interesting thoughts on man-made reasoning, the Google group made TensorFlow.

v. **Keras -** Keras is an undeniable level API to fabricate and prepare profound learning models. It's utilized for quick prototyping, high level exploration, and creation. Keras has a straightforward, predictable point of interaction upgraded for normal use cases. It gives clear and noteworthy criticism to client blunders. Keras models are made by interfacing configurable structure impedes together, with couple of limitations. Compose custom structure blocks to communicate novel thoughts for research. Make new layers, misfortune works.

vi. **Matplotlib -** Matplotlib is one of the most famous Python bundles utilized for information perception. It is a cross-stage library for making 2D plots from information in clusters. It gives an article arranged API that aides in implanting plots in applications utilizing Python GUI toolboxs.

vii. **Sys -** The sys module in Python gives different capacities and factors. It permits working on the mediator as it gives admittance to the factors and capacities that collaborate unequivocally with the translator.

viii. **Torch (Version - '1.10.0') -** Torch is an open-source AI library, a logical processing system, and a content language in view of the Lua programming language.

ix. **Pytorch -** PyTorch is an open source AI library for Python and is totally founded on Torch. It is essentially utilized for applications, for example, regular language handling. PyTorch is created by Facebook's man-made reasoning exploration bunch alongside Uber's "Pyro" programming for the idea of in-fabricated probabilistic programming. Normal Origin. At first, Torch was created and later, PyTorch was created as a Python execution of Torch. The two structures have been created by Facebook. Both are open source.

x. **Pprint -** The python module pprint is utilized for giving legitimate printing organizations to different information objects in python. Those information items can address a word reference information type or even an information object containing the JSON data.

xi. **Os -** The OS module in Python gives capacities for connecting the working framework.

xii. **Random -** The random module is used to generate random values.

xiii. **JSON -** JSON JavaScript Object Notation is a configuration for organizing information. We should see a basic model where we convert the JSON objects to Python items as well as the other way around.

xiv. **Albumentation -** Albumentations is a library of python for quick & adaptable picture expansions. Albumentations proficiently executes a vast assortment of picture change tasks that are enhanced for execution, and does as such while giving a compact, yet strong picture expansion interface for various PC vision undertakings, including object characterization, division, and discovery.

xv. **Glob -** In Python, we have many in-fabricated modules for performing different undertakings, and one of such errands we need to perform with the Python modules is finding and finding every one of the documents present in our framework, which follows a comparative example. This comparative example can be a record expansion, the document name's prefix, or any comparability between two or many records. We have a wide range of Python modules with which we can undoubtedly play out this errand utilizing a Python program, however not every one of the modules are basically as productive as others. In this instructional exercise, we will find out around one of such proficient modules, i.e., glob module in Python, with which we can perform document coordinating with a particular example by utilizing it inside a program. We will learn exhaustively about the glob module in Python, how we can utilize it inside a program, what its key elements are and the utilization.

xvi. **Shutil -** This module helps in mechanizing the most common way of duplicating and expulsion of records and registries. shutil.copy() technique in Python is utilized to duplicate the substance of source document to objective record or registry. It additionally protects the document's authorization mode however other metadata of the record like the record's creation and adjustment times isn't safeguarded.

3.3.2 Hardware Requirement - Laptops that are smoothly working, RAM (8Gb min), GPU

4. Project Methodology

4.1 System Design

The model performance is being improved with the removing the class imbalance problem using various Augmentation approaches by building a custom Augmented Dataset generator & a Custom Augmentation library. The parameters on which the techniques will be compared are on accuracy to solve the Class imbalance problem to maximize accuracy and reduce error & find the best possible method to solve it. System design shows the overall design of the system.

4.2 System Diagram

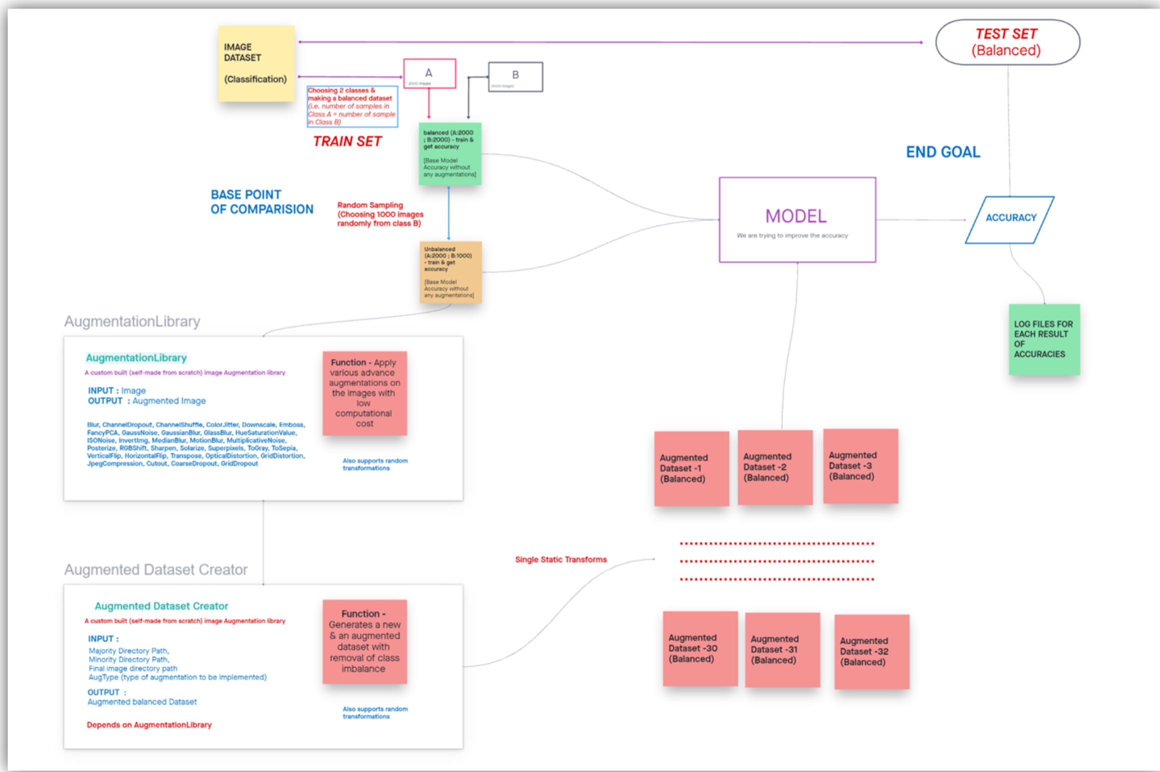


Figure.25 System Diagram

Initially we are taking a classification image data set. It has a train and test set. The trained model is trained, it is tested upon the test set and we are storing the accuracies in a log file for our records.

We are choosing two classes out of it for balancing the dataset i.e. making the number of samples in class A equal to number of sample in class B

We are passing it through the model and testing it with the test set and we are getting the accuracy.

We are creating an imbalance by choosing some random samples from class B and making an unbalanced data set which has the same number of images in class A and smaller images sampled in class B. We are passing it through the model and finding and storing in the accuracy

We are applying an augmentation technique to remove the class imbalance problem. We are making a new augmented image in which it has undergone some transformations like we can rotate or flip or increase the brightness or increase the contrast, etc.

So, different types of augmentation techniques are applied to make a new image so that the model could learn even if there are fewer samples in a class compared to others or if the sample dataset has less number of samples. We are making new samples using the existing samples. This helps the model to learn better even if there is less data available.

We are applying various augmentation techniques on the unbalanced data set and we generate different augmented balanced data sets. We are passing it through the model, training the model, finding the test accuracy & storing it in a log text file for our record.

Based on the accuracies of all types of augmented balanced datasets, we evaluate the augmentation techniques.

In the custom built augmentation library takes an input image, applies advanced augmentation techniques on these images and returns an augmented image. The library is better than others as it has low computational cost unlike in tensorflow or pytorch augmentation libraries that convert images to a tensor, applying the transformation on the tensor and again changing back the array into image. So, To reduce this, we directly applied the transformation on the image array and returned an augmented image. It supports many types of augmentations as follows:

- Blur - Blur the input image using a random-sized kernel.
- CLAHE - Apply Contrast Limited Adaptive Histogram Equalization to the input image.
- ChannelDropout - Drops out a channel based on a range
- ChannelShuffle - Randomly rearrange channels of the input RGB image.
- ColorJitter - Randomly changes the brightness, contrast, and saturation of an image.
- Downscale - Decreases image quality by downscaling and upscaling back.
- Emboss - Emboss the input image and overlays the result with the original image.
- FancyPCA - Augment RGB image using FancyPCA from Krizhevsky's paper "ImageNet Classification with Deep Convolutional Neural Networks"
- GaussNoise - Apply gaussian noise to the input image.
- GridDistortion - Grid-distortion is an image warping technique which is driven by the mapping between equivalent families of curves, arranged in a grid structure. Until recently only curve sets arranged in a regular rectangular grid were considered.
- JpegCompression - Decrease Jpeg compression of an image.
- Cutout - CoarseDropout of the square regions in the image
- CoarseDropout - CoarseDropout of the rectangular regions in the image
- GridDropout - GridDropout, drops out rectangular regions of an image and the corresponding mask in a grid fashion.

- GaussianBlur - Blur the input image using a Gaussian filter with a random kernel size.
- GlassBlur - Apply glass noise to
- the input image.
- HueSaturationValue - Randomly change hue, saturation and value of the input image.
- ISONoise - Apply camera sensor noise.
- InvertImg - Invert the input image by subtracting pixel values from 255.
- MedianBlur - Blur the input image using a median filter with a random aperture linear size.
- MotionBlur - Apply motion blur to the input image using a random-sized kernel.
- MultiplicativeNoise - Multiply image to random number or array of numbers.
- Posterize - Reduce the number of bits for each color channel.
- RGBShift - Randomly shift values for each channel of the input RGB image.
- Sharpen - Sharpen the input image and overlays the result with the original image.
- Solarize - Invert all pixel values above a threshold.
- Superpixels - Transform images partially /completely to their superpixel representation. This implementation uses skimage's version of the SLIC algorithm.
- ToGray - Convert the input RGB image to grayscale. If the mean pixel value for the resulting image is greater than 127, invert the resulting grayscale image.
- ToSepia - Applies sepia filter to the input RGB image
- VerticalFlip - Flip the input vertically around the x-axis.
- HorizontalFlip - Flip the input horizontally around the y-axis.
- Transpose - Transpose the input by swapping rows and columns.
- OpticalDistortion - Distortion can be thought of as the difference in magnification across a field of view. This is usually calculated as a percentage of image size. By taking the measured distance in the image and comparing it to the predicted distance, we can calculate the optical distortion

Figure.26 Augmentation techniques

The augmented data set creator uses an augmentation library that generates a new augmented balanced dataset. The unbalanced data set that we gave as an input is balanced by applying various augmentation techniques with the existing samples. The model is trained with these datasets and the test accuracy is stored for our evaluation.

4.3. Extensions made on the Existing Model

- Compared many Augmentation libraries, and chosen the best one based on various parameters
- Built a new Augmentation library from scratch - Static Aug Library & Random Aug Library
- Unit-Tested the Augmentation library using Unit-Test Library to make it efficient. Testing Static Aug Library & Testing Random Aug Library
- Created a detailed summary of each type of augmentations that the library supports
- Developed a Dataset Balance & Unbalance Creator
- Built a Augmentation Dataset Creator based on the Augmentation library
- Implemented the RESNET18 model [pre-trained = False]
- Added layers to get the accuracy metrics – train & test accuracies
- Ran for 100 epochs for each test case [Augmented Dataset] & choosing the best trained model using the test accuracies

- Added the logging code to store the log file for every experiment
- Ran the test cases for 1:2, 1:4, 1:8, 1:16, 1:20 imbalance
- Ran the tests for both static & Random Augmentations
- Ran the tests for 3 datasets
- Analyzed & compared the results to compare the augmentations that improve the model accuracy

5. MODEL ARCHITECTURE & SUMMARY

5.1. Model Architecture

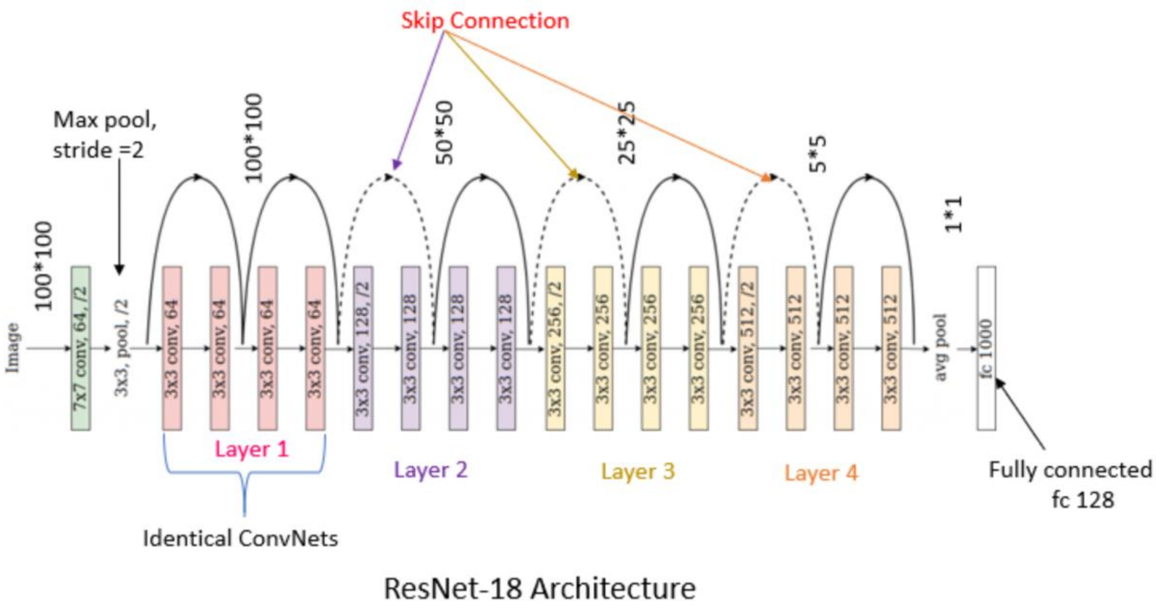


Figure.27 ResNet-18 Architecture

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

:ures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of block

Figure.28 Building blocks

- X is input, Y is yield , Y=F(X). The rationale behind RESNET is to make Input=Output
- In the event that we make F(X)=0, it is simple as far as we're concerned to make input = yield

  Y=F(X)+X

  Y=X+0

  Y=X
- In ordinary organizations we gain from Y however in Residual organization we gain from F(X) and our objective is to create F(X)=0 then no one but we can make input=output
- RESNET first presented the idea of skip association.
- Here, we add the first contribution to the result of the convolutional block
- It has 18 layers

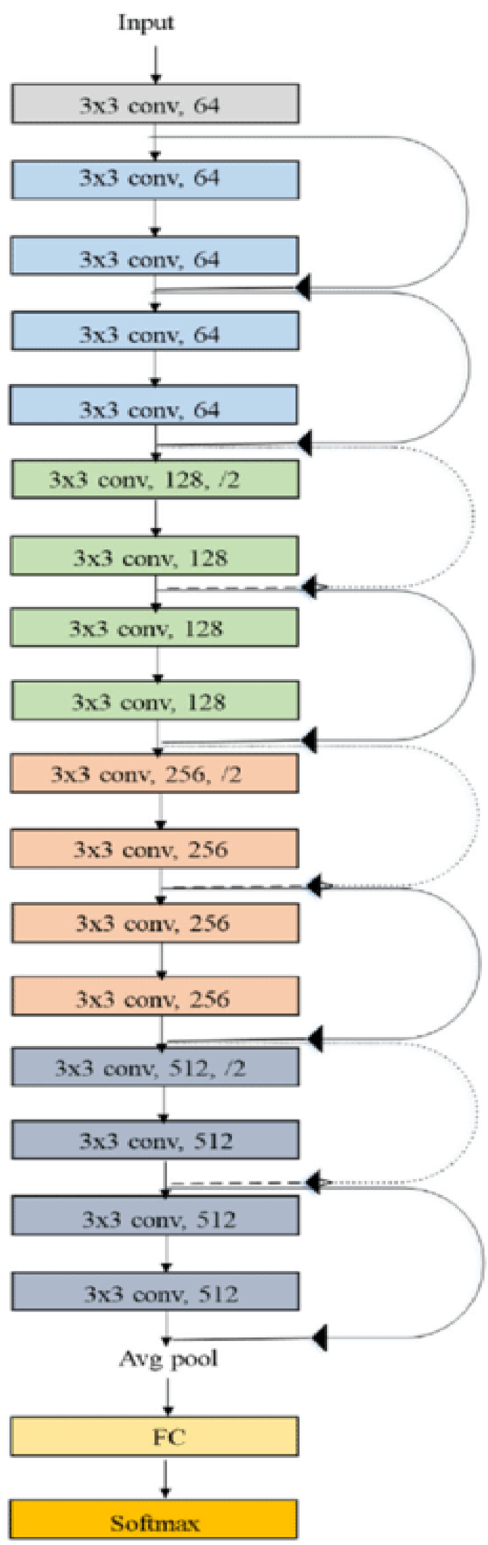


Figure.29 ResNet-18 layers

5.2. Model summary

```
ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
  (layer1): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (1): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (layer2): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
```

```
 (relu): ReLU(inplace=True)
 (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
 (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
 (downsample): Sequential(
  (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
  (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
 )
)
(1): BasicBlock(
 (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
 (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
 (relu): ReLU(inplace=True)
 (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
 (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)

ayer3): Sequential(
(0): BasicBlock(
 (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
 (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
 (relu): ReLU(inplace=True)
 (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
 (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
 (downsample): Sequential(
  (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
```

```
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (1): BasicBlock(
    (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(layer4): Sequential(
  (0): BasicBlock(
    (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (downsample): Sequential(
      (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (1): BasicBlock(
    (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
```

```
    (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
   )
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
  (fc): Sequential(
   (0): Linear(in_features=512, out_features=512, bias=True)
   (1): ReLU()
   (2): Dropout(p=0.2, inplace=False)
   (3): Linear(in_features=512, out_features=2, bias=True)
   (4): LogSoftmax(dim=1)
  )
)
```

Figure.30 ResNet-18 Model Architecture

6. EXPERIMENTATION AND RESULTS

This paper aims to improve the model performance while removing the class imbalance problem using various augmented balanced datasets. They are generated using a custom-built balanced dataset generator based upon the AugStatic library. AugStatic is a custom-built image augmentation library with lower computation costs than other image augmentation libraries. The augmented balanced datasets are inputted to the RESNET18 model. The best test accuracies are found for various class imbalance ratios for three classification datasets - Intel Scene, CIFAR10, and Chest X-Ray Pneumonia dataset, for 100 epochs, stored and analyzed. The research analysis has brought up some astounding insights into the model performance for various augmentation techniques.

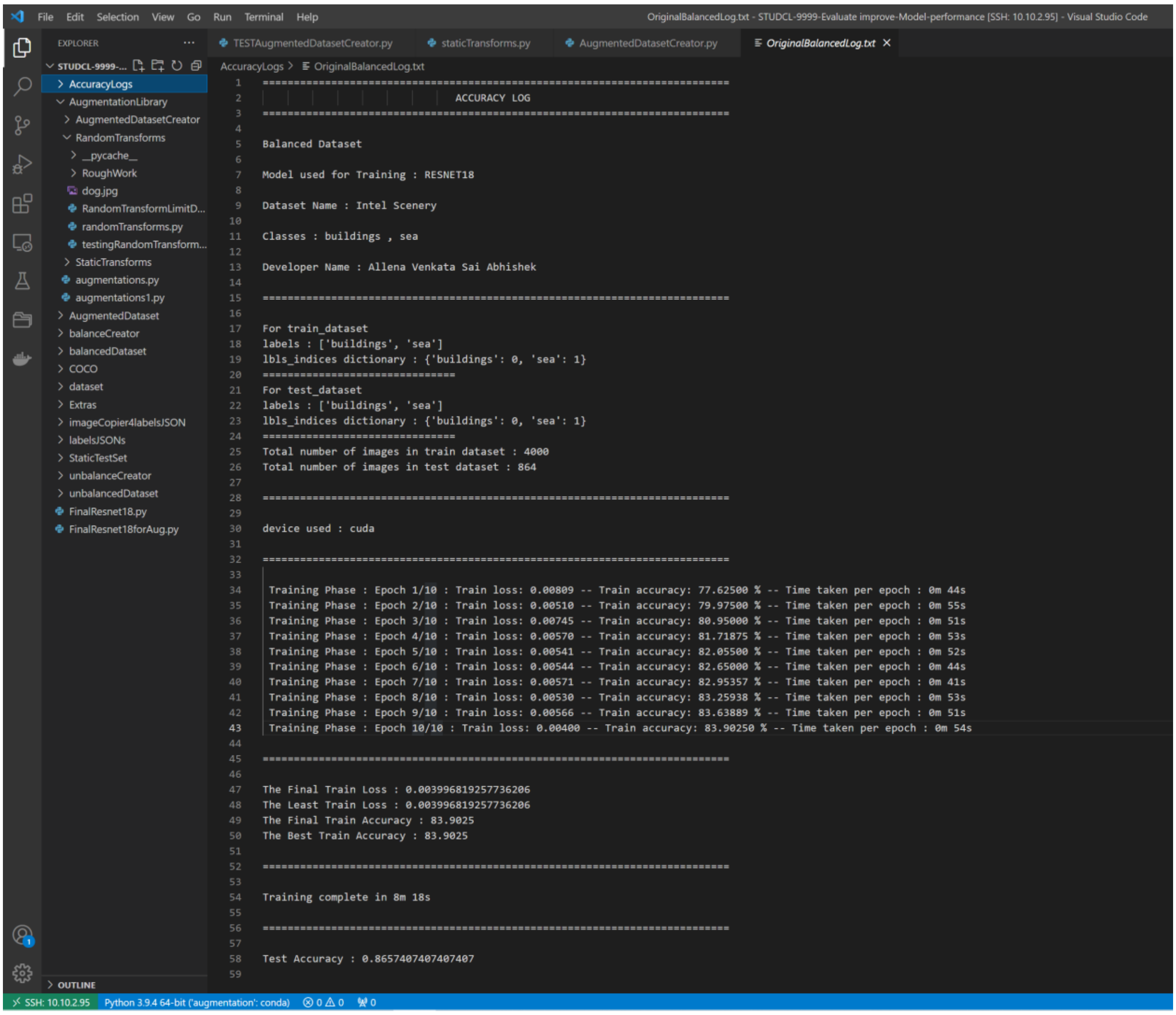


Figure.31 Output

It emphasizes to solve class imbalance problems using Random Sampling and advanced data augmentation techniques. It included the comparison of many augmentation libraries. It chose the best one based on various parameters, built a new Augmentation library, and analyzed each type of augmentation that the library supports. We balanced the two classes of original datasets by developing a Dataset Balance & created an imbalance randomly by using an unbalance creator. An augmentation Dataset Creator was built that used the Augmentation library.

The datasets generated are being used to train the RESNET18 model with additional layers to get the accuracy metrics – train & test accuracies. The model is being run for three datasets with hundred epochs for each class imbalance and we choose the best-trained model using the test accuracies found with the trained model that is being tested upon the fixed test set. This test has been done for various test cases that includes class imbalance ratios of 1:2, 1:4, 1:8, 1:16, and 1:20. These tests had been run for three datasets. The results had been analyzed & collated for different augmentation methods and three datasets. The resultant augmentations had been jotted down that improved the model accuracy. Augmentation techniques in amplyfying the accuracy of the model by disarding imbalance –

A. For Intel Scene Dataset –

INEFFECTIVE	**EFFECTIVE**
GaussianBlur ChannelDropout GridDropout Sharpen Solarize	FancyPCA GaussNoise InvertImg RGBShift HueSaturationValue JpegCompression MultiplicativeNoise

Table 3: Effective & Non-Effective Augmentations for Intel Scene Dataset

INVERSE TO IMBALANCE	**CONSTANT TO IMBALANCE**	**DIRECT TO IMBALANCE**
CLAHE Blur CoarseDropout	Median Blur Glass Blur Motion Blur	Cutout FancyPCA Posterize

	To Sepia Emboss InvertImg GaussNoise RGBShift ISONoise HorizontalFlip ColorJitter OpticalDistortion MotionBlur MultiplicativeNoise GridDistortion Superpixels Downscale Superpixels ToGray	HueSaturationValue JpegCompression ChannelShuffle VerticalFlip Transpose

Table 4: Variation of Augmentations with imbalance ratio for Intel Scene Dataset

B. For Chest Xray Pneumonia Dataset –

INEFFECTIVE	**EFFECTIVE**
GaussianBlur Glass Blur ChannelDropout GridDropout Sharpen Solarize	VerticalFlip Transpose GaussNoise MotionBlur HueSaturationValue

Table 5: Effective & Non-Effective Augmentations for Chest Xray Pneumonia Dataset

INVERSE TO IMBALANCE	**CONSTANT TO IMBALANCE**	**DIRECT TO IMBALANCE**
CLAHE Blur CoarseDropout	Median Blur Glass Blur Motion Blur To Sepia Emboss InvertImg GaussNoise RGBShift ISONoise HorizontalFlip ColorJitter OpticalDistortion MotionBlur MultiplicativeNoise GridDistortion Superpixels Downscale Superpixels ToGray	Cutout FancyPCA Posterize HueSaturationValue JpegCompression ChannelShuffle VerticalFlip Transpose

Table 6: Variation of Augmentations with imbalance ratio for Chest Xray Pneumonia Dataset

C. For CIFAR 10 Dataset –

INEFFECTIVE	**EFFECTIVE**
GaussianBlur ChannelDropout GridDropout Sharpen Solarize	FancyPCA GaussNoise InvertImg RGBShift HueSaturationValue

	JpegCompression MultiplicativeNoise

Table 7: Effective & Non Effective Augmentations for CIFAR 10 Dataset

INVERSE TO IMBALANCE	**CONSTANT TO IMBALANCE**	**DIRECT TO IMBALANCE**
CLAHE Blur CoarseDropout	Median Blur Glass Blur Motion Blur To Sepia Emboss InvertImg GaussNoise RGBShift ISONoise HorizontalFlip ColorJitter OpticalDistortion MotionBlur MultiplicativeNoise GridDistortion Superpixels Downscale Superpixels ToGray	Cutout FancyPCA Posterize HueSaturationValue JpegCompression ChannelShuffle VerticalFlip Transpose

Table 8: Variation of Augmentations with imbalance ratio for CIFAR 10 Dataset

7. CONCLUSION AND RECOMMENDATION

A. Conclusion

This project proposes an approach for solving class imbalance problems using Random Sampling and advanced data augmentation techniques. It included the comparison of many augmentation libraries. It chose the best one based on various parameters, built a new Augmentation library, and analyzed each type of augmentation that the library supports. We balanced the two classes of original datasets by developing a Dataset Balance & created an imbalance randomly by using an unbalance creator. An augmentation Dataset Creator was built that used the Augmentation library. The datasets generated are used to train the RESNET18 model with additional layers to get the accuracy metrics – train & test accuracies. The model is run for 100 epochs for each test case & choosing the best-trained model using the test accuracies. This test was done for the test cases for 1:2, 1:4, 1:8, 1:16, and 1:20 imbalance. These tests were run for three datasets. The results were analyzed & compared for the various augmentation techniques &3 datasets. The resultant augmentations were recorded that improved the model accuracy. Data scientists & ML Engineers can use the system to improve the model to learn more with fewer data by implementing the effective augmentation for the datasets based on insights derived from this research.

B. Future Scope

Augmentation libraries and techniques have improved dramatically over the past decade with an exponential increase in data. Few scopes for the future are to implement additional features such as getting the Random Transformations with the type and parameter values it, implementing the Random Augmentations in Augmented Dataset Creator, Getting the Accuracies for Random Augmented Datasets, implementing more cases a combination by using multiple transforms and getting the prediction scores and using the combination of the majority, moderate and lower prediction scores for the augmentation techniques and then getting the accuracies of each for evaluation. GANs can also be implemented to create augmented images. Data scientists & ML Engineers can use the system to improve the model to learn more with fewer data by implementing the effective augmentation for the datasets based on insights derived from this research with further implementation of the research in data augmentation & machine learning. This will help the customers to enhance the business etc.

8. BIBLIOGRAPHY

[1] Mateusz Buda, Atsuto Maki, Maciej A. Mazurowski "A systematic study of the class imbalance problem in convolutional neural networks" arXiv:1710.05381v2 [cs.CV] Oct 2018

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun "Deep Residual Learning for Image Recognition" arXiv:1512.03385v1 [cs.CV] Dec 2015

[3] Connor Taghi M. Khoshgoftaar "A survey on Image Data Augmentation for Deep Learning" Shorten and Khoshgoftaar J Big Data (2019)

[4] Michał Koziarski " Radial-Based Undersampling for Imbalanced Data Classification" Michał Koziarski arXiv:1906.00452v2 [cs.LG] Apr 2021

[5] Marcus D. Bloice, Christof Stocker, Andreas Holzinger "Augmentor: An Image Augmentation Library for Machine Learning" arXiv:1708.04680v1 [cs.CV] Aug 2017

[6]. Alexandr A. Kalinin, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Alexander Buslaev "Albumentations: fast and flexible image Augmentations" Feb 2020

[7] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, David Lopez-Paz "MixUp augmentation for image classification"arXiv:1710.09412v2 [cs.LG] Apr 2018

[8] Zhiting Hu, Bowen Tan, Ruslan Salakhutdinov, Tom Mitchell, Eric P. Xing "Learning Data Manipulation for Augmentation and Weighting " arXiv:1910.12795v1 [cs.LG] Oct 2019

[9] Shanchuan Lin, Linjie Yang, Imran Saleemi, Soumyadip Sengupta "Robust High-Resolution Video Matting with Temporal Guidance" arXiv:2108.11515v1 [cs.CV] Aug 2021

[10] Xu Sun, Huihui Fang, Yehui Yang, Dongwei Zhu, Lei Wang, Junwei Liu, Yanwu Xu "Robust Retinal Vessel Segmentation from a Data Augmentation Perspective" arXiv:2007.15883v2 [eess.IV] Sep 2021

[11] Wanwan Zheng, Mingzhe Jin "The Effects of Class Imbalance and Training Data Size on Classifier Learning" Springer Feb 2020

[12] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton "ImageNet Classification with Deep Convolutional Neural Networks " NeurIPS Proceedings 2012

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio “Generative Adversarial Networks “ arXiv:1406.2661v1 [stat.ML] Jun 2014

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer “SMOTE: Synthetic Minority Over-sampling Technique” Journal of Artificial Intelligence Research 16 (2002)

[15] Elena Limonova, Daniil Alfonso, Dmitry Nikolaev, Vladimir V. Arlazarov “ResNet-like Architecture with Low Hardware Requirements” arXiv:2009.07190v2 [cs.CV] Oct 2020

[16] Anusha Kanukolanu, Dr. S Phani Kumar, Allena Venkata Sai Abhishek (2022) “Augmentation Techniques Analysis with removal of Class Imbalance using PyTorch for Intel Scene Dataset” https://ijirt.org/Article?manuscript=155039

[17] Allena Venkata Sai Abhishek, Dr. Venkateswara Rao Gurrala, "AUGMENTED BALANCED IMAGE DATASET GENERATOR USING AUGSTATIC LIBRARY", IJRAR - International Journal of Research and Analytical Reviews (IJRAR), E-ISSN 2348-1269, P- ISSN 2349-5138, Volume.9, Issue 2, Page No pp.536-543, May 2022, Available at : http://www.ijrar.org/IJRAR22B1901.pdf

[18] Allena Venkata Sai Abhishek , Sonali Kotni, 2021, Detectron2 Object Detection & Manipulating Images using Cartoonization, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 08 (August 2021)

[19] Allena Venkata Sai Abhishek, Venkateswara Rao Gurrala "AugStatic - A Light-Weight Image Augmentation Library", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN: 2349-5162, Vol.9, Issue 5, page no.b735-b742, May-2022, Available :http://www.jetir.org/papers/JETIR2205199.pdf

[20] Allena Venkata Sai Abhishek, Dr. Venkateswara Rao Gurrala, Dr. Laxman Sahoo, "RESNET18 MODEL WITH SEQUENTIAL LAYER FOR COMPUTING ACCURACY ON IMAGE CLASSIFICATION DATASET", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.10, Issue 5, pp.c176-c181, May 2022, Available at :http://www.ijcrt.org/papers/IJCRT2205235.pdf

[21] Allena Venkata Sai Abhishek and Venkateswara Rao Gurrala, Improving Model Performance and Removing the Class Imbalance Problem Using Augmentation, International Journal of Advanced Research in Engineering and Technology (IJARET). 13(5), 2022, pp. 14-22 doi: https://doi.org/10.17605/OSF.IO/CQMY5