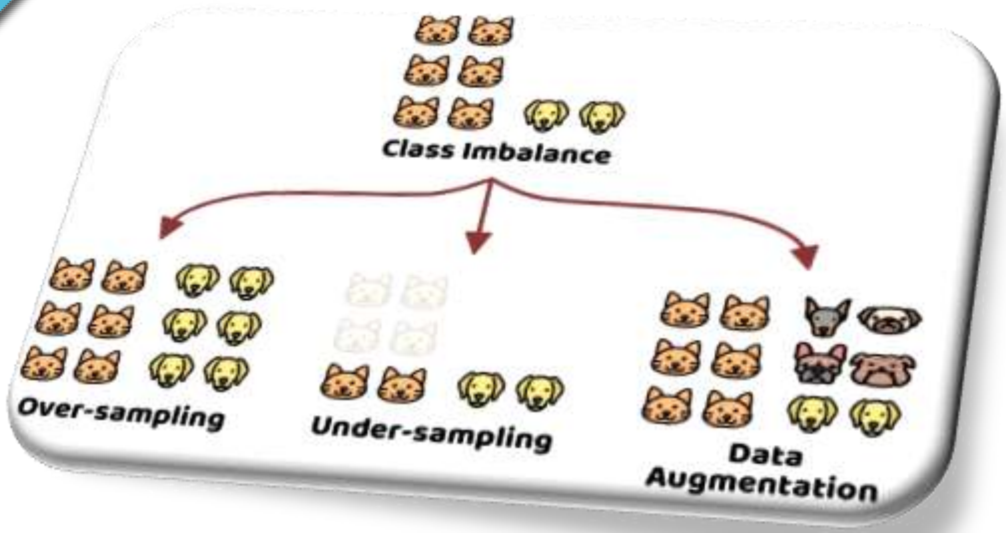


IMPROVING MODEL PERFORMANCE & REMOVING THE CLASS IMBALANCE PROBLEM USING AUGMENTATION

Allena Venkata Sai Abhishek

MTech Data Science

GITAM University, Visakhapatnam, India

PROJECT REVIEW- 2020-2022 ADMITTED BATCH
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
ALLENA VENKATA SAI ABHISHEK -122021601009

What is Class Imbalance Problem?

- Unequally distributed (Major & Minor Class)

Types of Class Imbalance

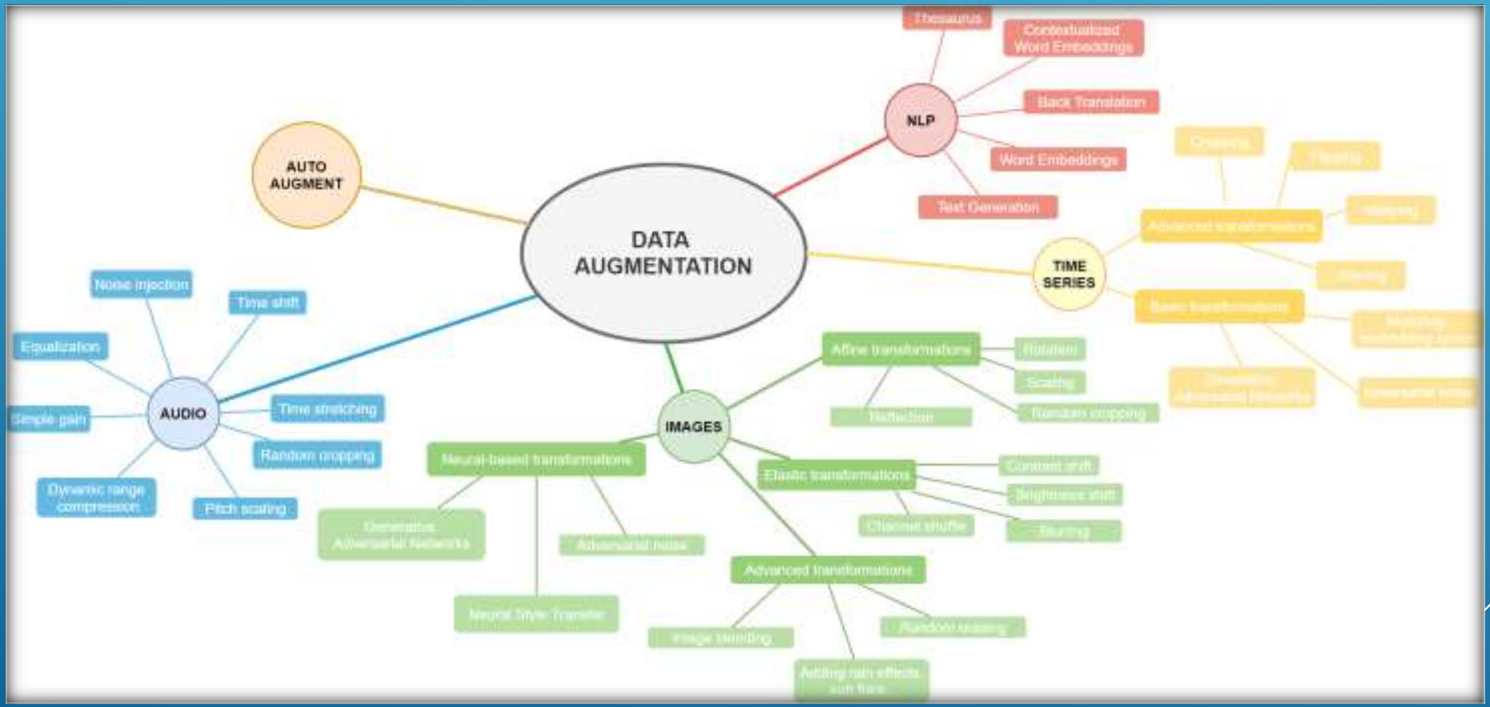
- Slight Imbalance (<1:10)
- Severe Imbalance (>1:10)

Solution for removing Class Imbalance Problem

- Over-Sampling
- Under-Sampling

What is Augmentation?

- Data augmentation is a technique to artificially create new training data from existing training data.
- Data augmentation is a strategy that enables to significantly increase the diversity of data available for training models, without actually collecting new data.



Motivation Behind the Topic

The class imbalance problem is a painful feature in real world data

Many augmentations supported by various augmentation libraries

LITERATURE SURVEY

	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 Classifer Learning	naïve Bayes, logistic regression model are 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 that allows for fine-grained control over the creation of augmented data and provides many functions for augmentation technique.

	Journal Type and year	Authors	Title	Summary
6	IEEE, 2018	Alexandr A. Kalinin, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Alexander Buslaev	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 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 downsampled 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 upsampling. Then DGF is used to upsample 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.)

Click me for some more info

Also in next page too

Abstract

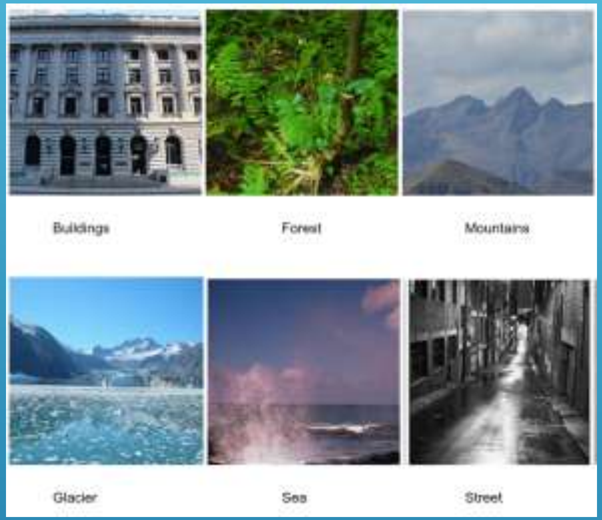
- One particularly painful feature of real-world data is that it can be imbalanced. An imbalanced dataset is a dataset where there are many more datapoints for one category than others.
- This research presents the Solving Class Imbalance problems using Random Sampling & Data Augmentation Techniques. Readers will understand how Under-Sampling, Over-Sampling, Synthetic Minority Over-sampling Technique & Data Augmentation using Image and custom datasets.
- 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.

Proposed System

- **Dataset :** An Image Classification Dataset (Intel Scenery), Chest Xray Pneumonia, CIFAR 10
- **Model :** Resnet18
- **Solution :**

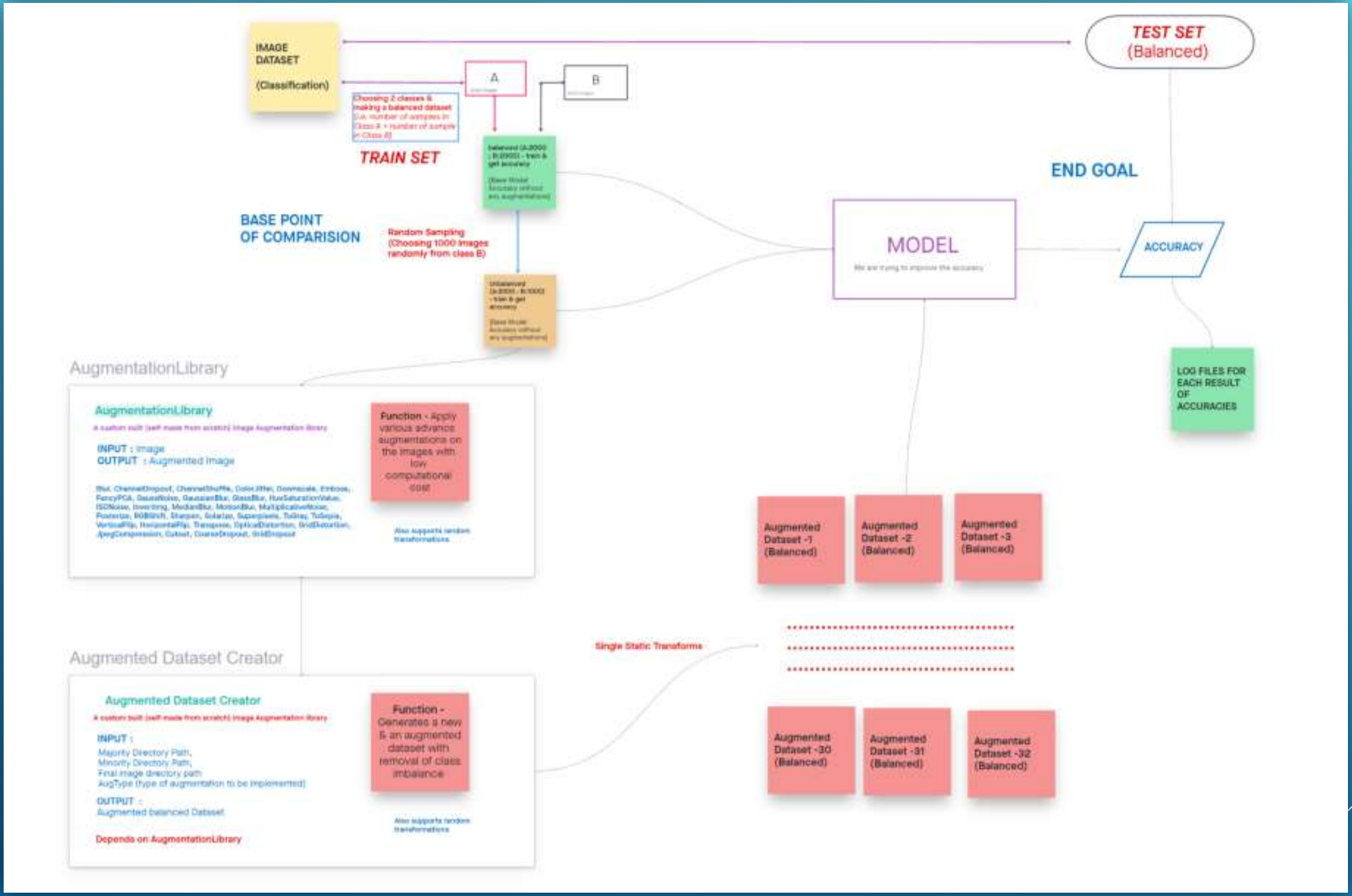
  Techniques implemented to solve class imbalance -

  - **Under Sampling –**
    - Random Under Sampling
  - **Over Sampling –**
    - Random Over Sampling,
    - Data Augmentation



- **Metric used to Compare :** Accuracy
- **Goal :** Find the best Data augmentation technique for the dataset with construction of augmentation library & augmented dataset generator

Workflow of the Proposed System



A Video Explanation of the workflow

AUGMENTATION LIBRARIES

Package Name	imgaug package	Albumentations package	Augmentor package
Description	It contains: Over 40 image augmentors and augmentation techniques;	Over 60 pixel level and spatial level transformations	Augmentor package contains less possible augmentations than other packages
	Functionality to augment images with masks, bounding boxes, keypoints and heatmaps.	Transforming images with masks bounding boxes and keypoints;	Extra features like size-preserving rotations, size-preseving shearing and cropping, which is more suitable for machine learning.
	This functionality makes it very easy to augment the dataset containing images for segmentation and object detection problems	**Organizing augmentations into pipelines -** Albumentations package allows to construct advanced pipelines similar to imgaug pipelines.	Augmentor package also allows to compose augmentation pipelines and use them with PyTorch.
	Complex augmentation pipelines	**PyTorch integration** - effortlessly migrate from torchvision to Albumentations, because this package provides specialized utilities to be used with PyTorch. Migrating to Albumentations will help to speed up the data generation part and train deep learning models faster. See detailed tutorial on migration from torchvision to Albumentations	
	Many helper functions for augmentation visualization, convertion, and more.	Albumentations is a fast and flexible image augmentation library.	

Augmentation Package Detailed Comparision .XLSX

docs.google.com/spreadsheets/d/1MxAh7mP-Q2ZqWTU61KtV-OvOqPlcTnnF/edit#gid=1374483179

Last edit was on November 24, 2021

Augmentation Packages Detailed

Augmentation Techniques

- 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.

- 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.

- 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

Augmentation Dataset Creator

Augmented Dataset Creator

Augmented Dataset Creator

A custom built (self-made from scratch) image Augmentation library

INPUT :

Majority Directory Path,
Minority Directory Path,
Final image directory path
AugType (type of augmentation to be implemented)

OUTPUT :

Augmented balanced Dataset

Depends on AugmentationLibrary

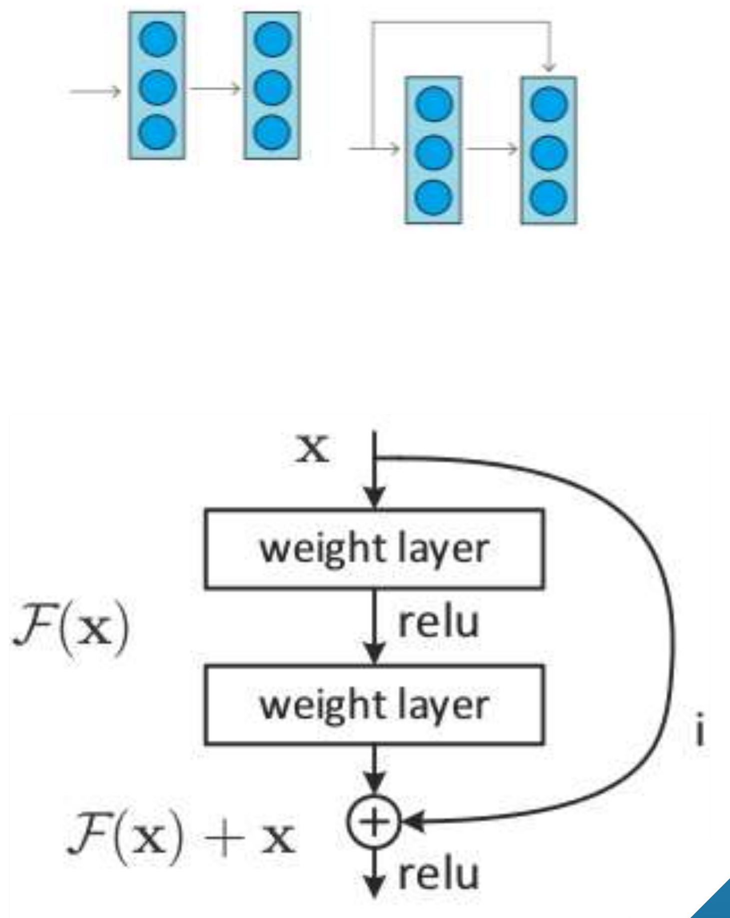
Function -
Generates a new & an augmented dataset with removal of class imbalance

Also supports random transformations

MODEL USED & ARCHITECTURE

RESNET18

- X is input, Y is output , Y=F(X). The logic behind RESNET is to make Input=Output
- If we make F(X)=0, then it is easy for us to make input = output
  Y=F(X)+X
  Y=X+0
  Y=X
- In normal networks we learn from Y but in Residual network we learn from F(X) and our target is to make F(X)=0 then only we can make input=output
- RESNET 1st introduced the concept of skip connection.
- Here, we add the original input to the output of the convolutional block
- It has 18 layers



SKIP CONNECTION

Why is the relu applied after adding the skip connection?

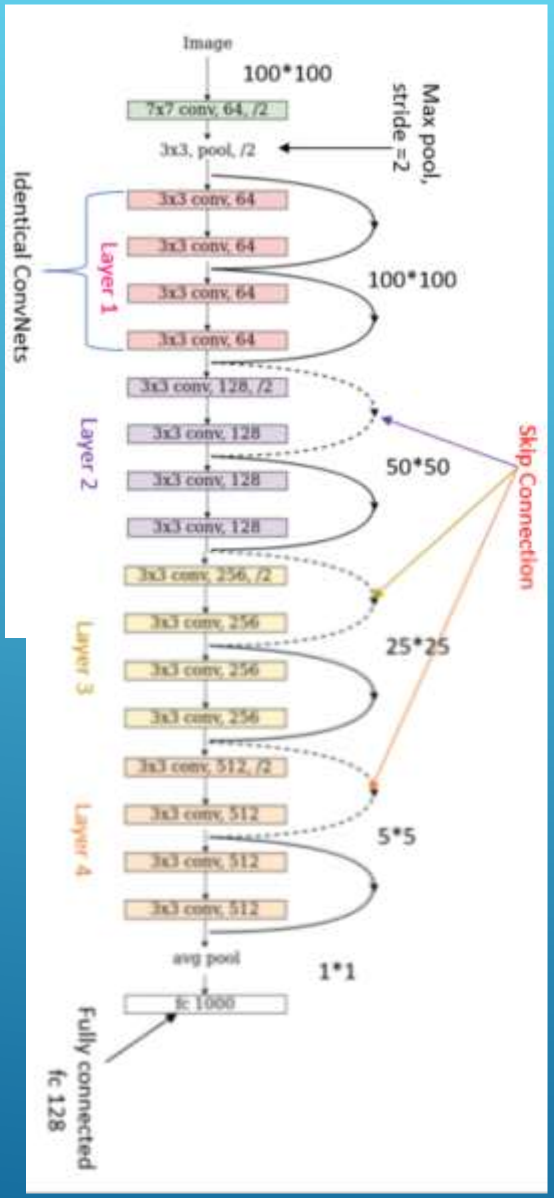
- ❑ If we had performed relu before addition, then the residues will all be positives or zero. Only positive increments to the identity are learnt, which significantly reduces the learning capacity.
- ❑ For example in the sin function, sin(3π/2) = -1, which would need negative residue.
- ❑ Similarly, using sigmoid will also be disadvantageous, because it produces residues only within 0 to 1.

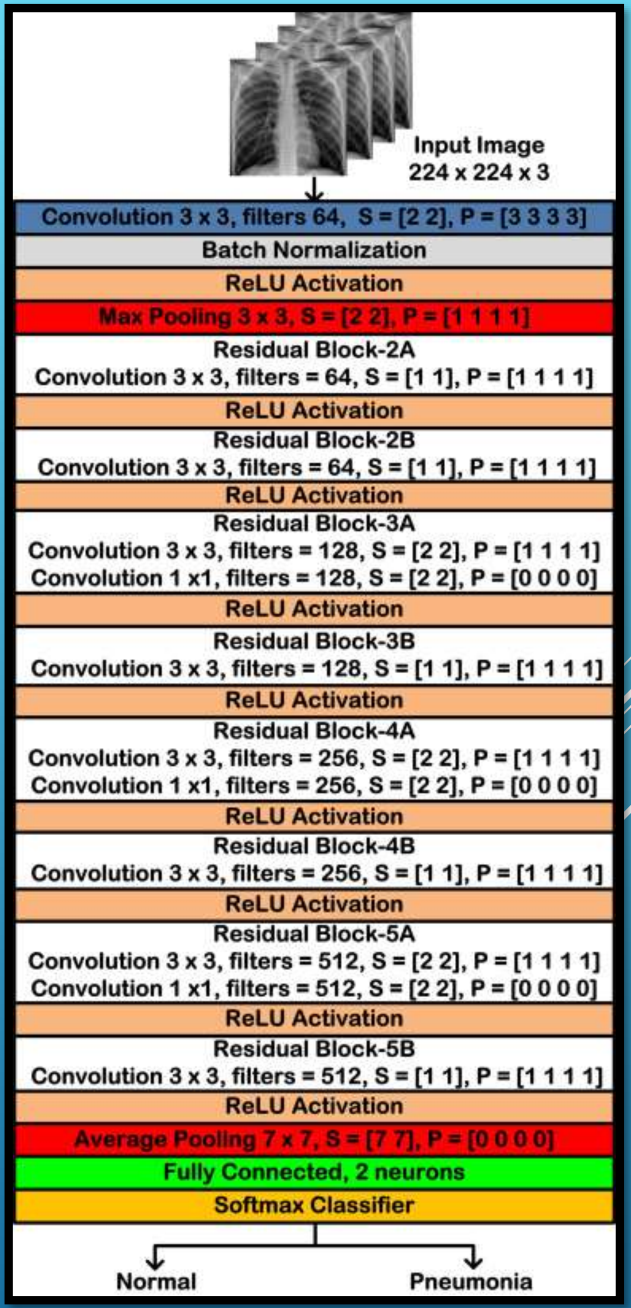
==

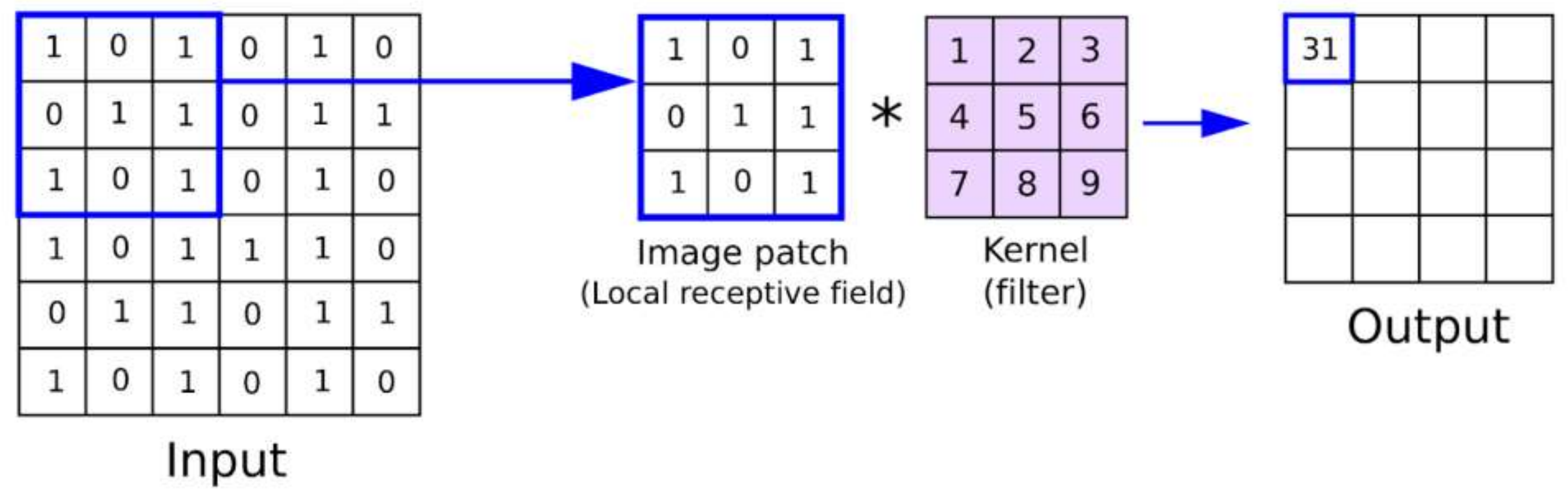
Why are there two weight layers in one residual block?

- ❑ if we had used a single weight layer, adding skip connection before relu, gives F(x) = Wx+x, which is a simple linear function.
- ❑ This is equivalent to just a single weight layer and there is no point in adding skip connection.
- ❑ So we need at least one non-linearity before adding skip connection, which is achieved by using two layers.

conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connecti





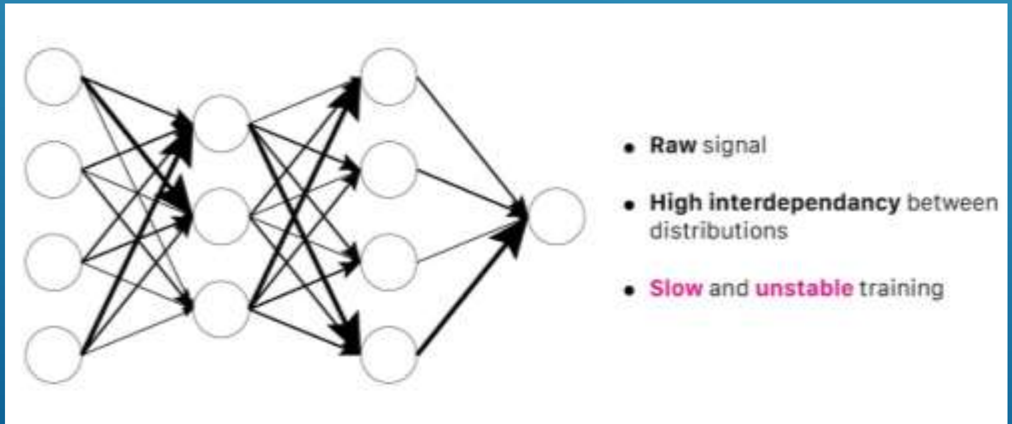


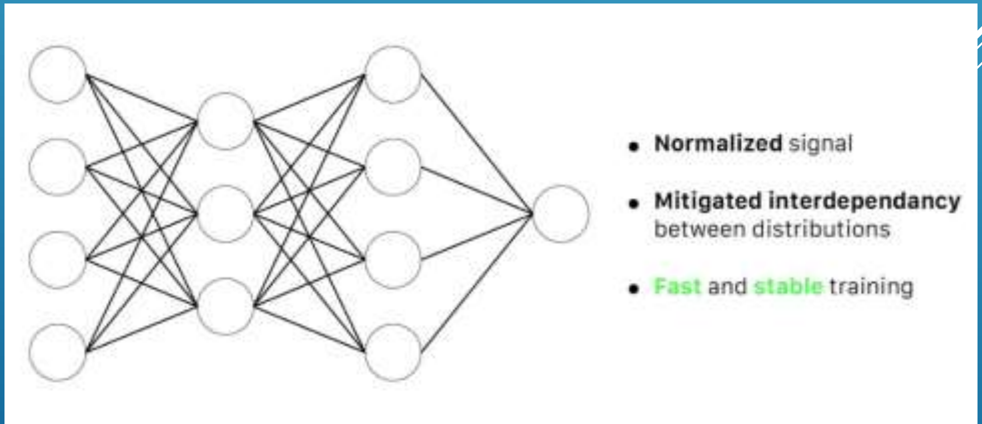
CONVOLUTION OPERATION

We do elementwise multiplication using a kernel on the input image & it is used for feature extraction.

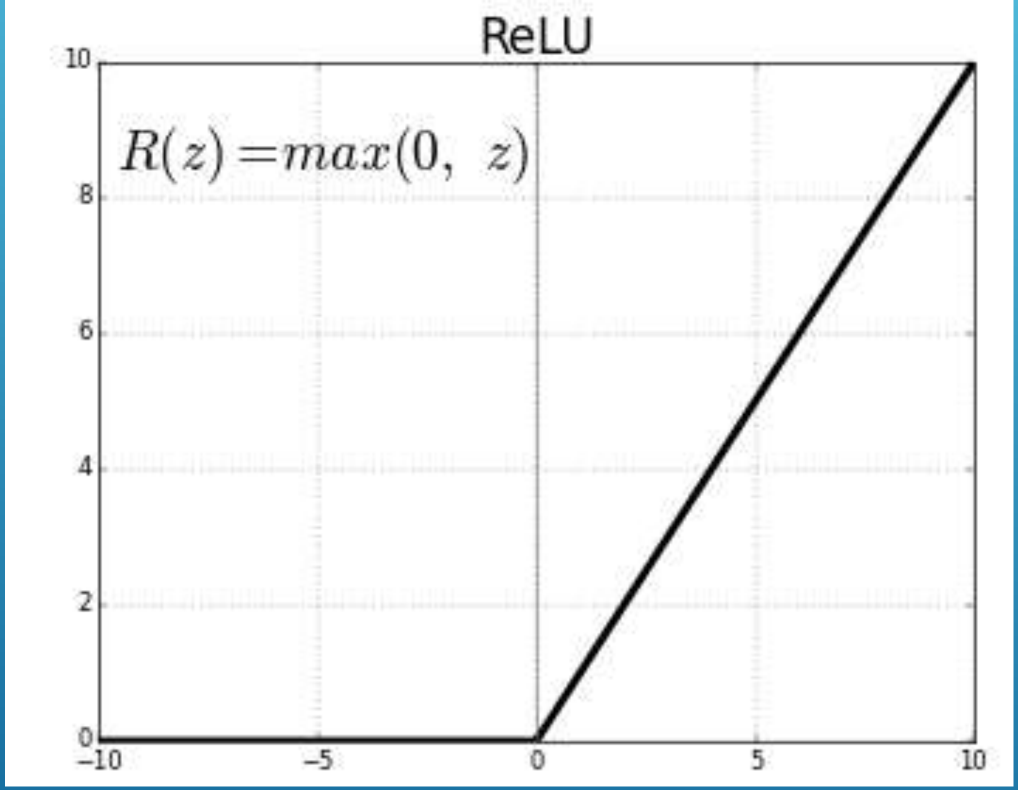
BATCH NORMALIZATION

- For training very deep neural networks
- the network to do learning more independently.
- normalize the output of the previous layers before passing them on as the input of the next hidden layer.
- Stabilizes the learning process
- Reduces the number of training epochs required to train deep networks.

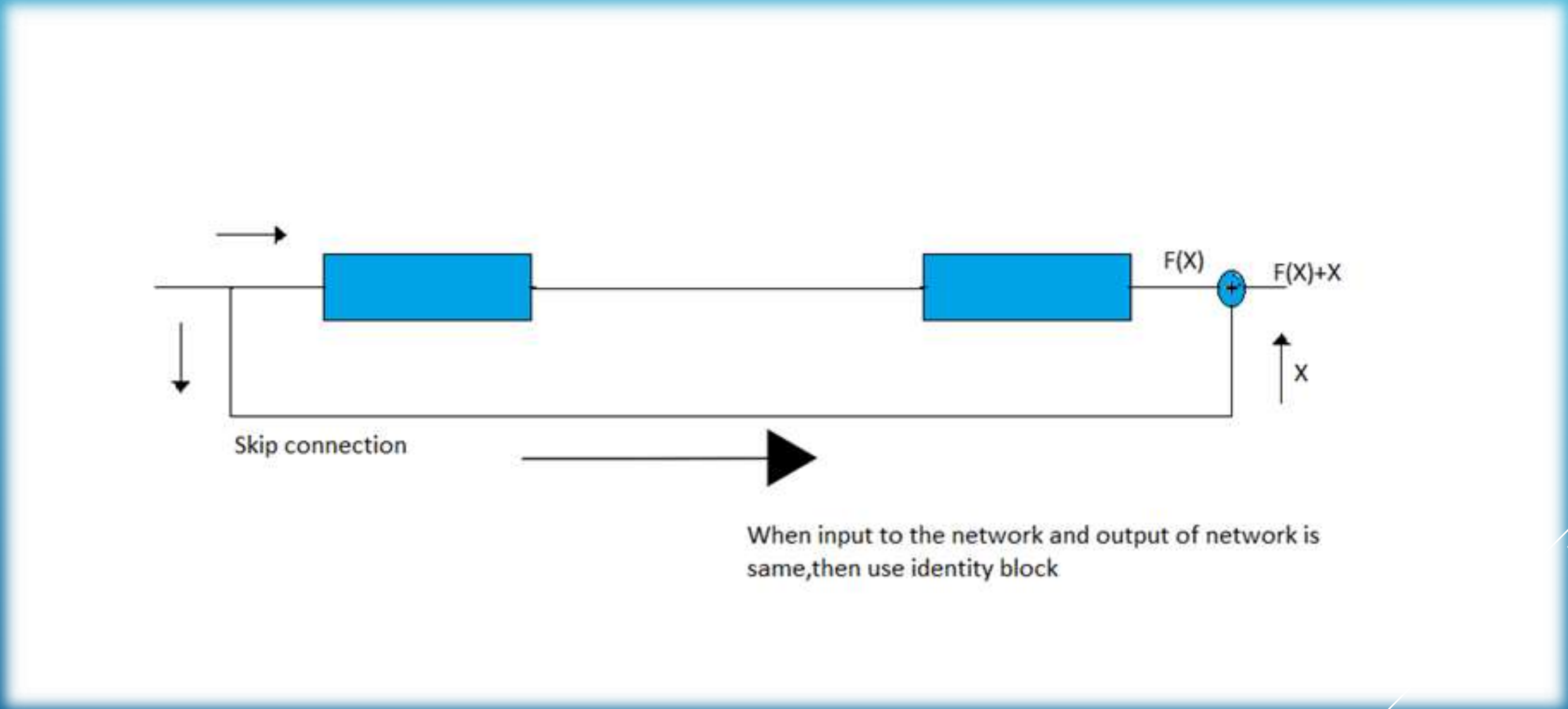




RELU ACTIVATION

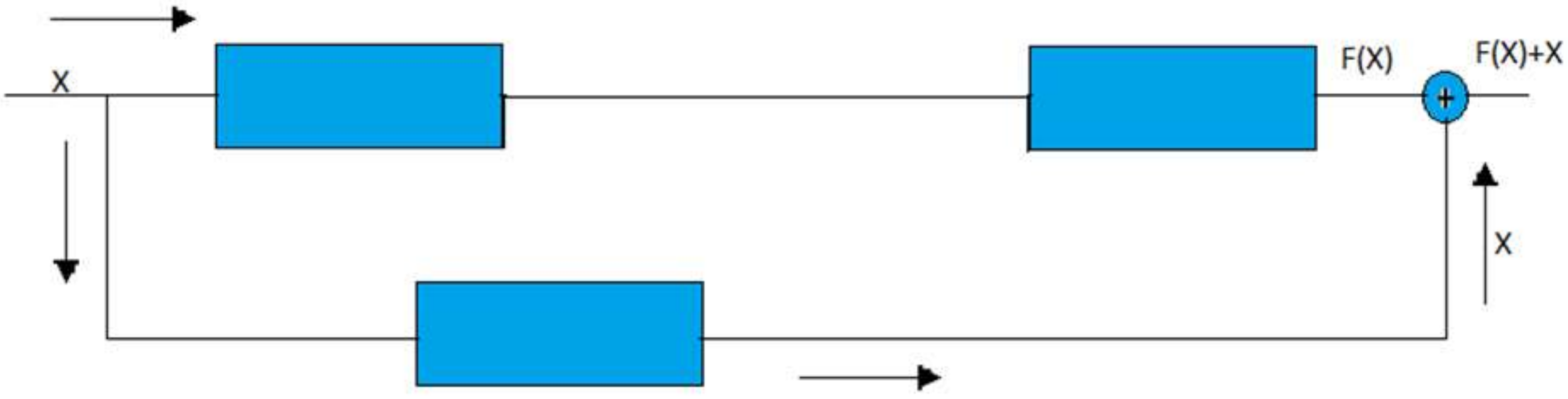


IDENTITY BLOCK (INPUT SIZE = OUTPUT SIZE)



CONVOLUTIONAL BLOCK

(INPUT SIZE != OUTPUT SIZE)



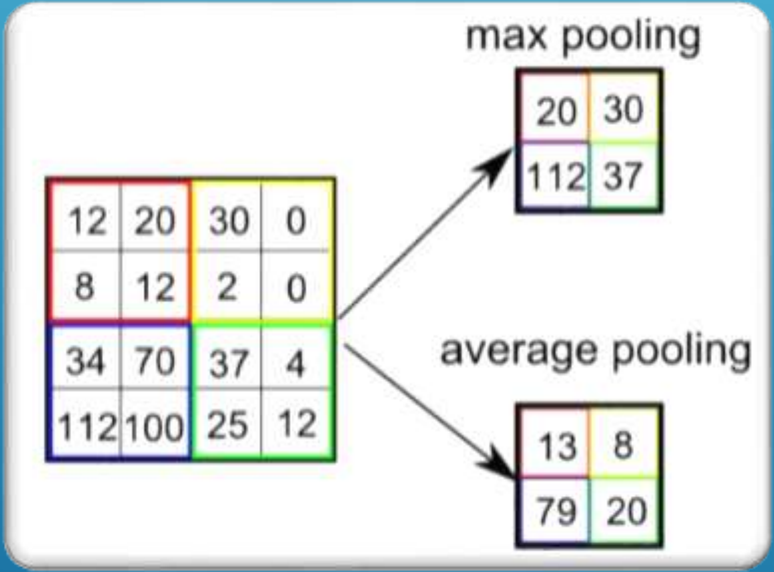
Convolutional layer added in shortcut path

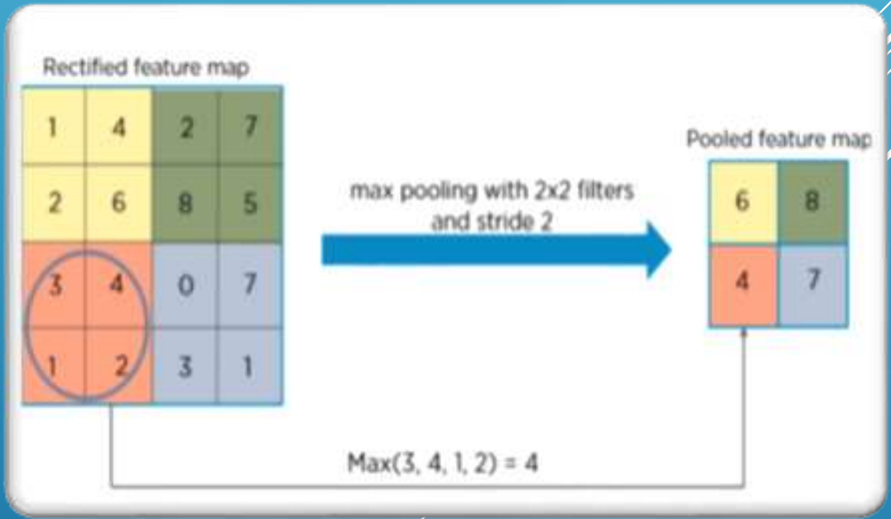
POOLING OPERATION

- ❑ Pooling reduces the size of the feature map in order to help us with using the fewer parameters in the network i.e dimensionality reduction.

- ❑Used to find the essence.

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



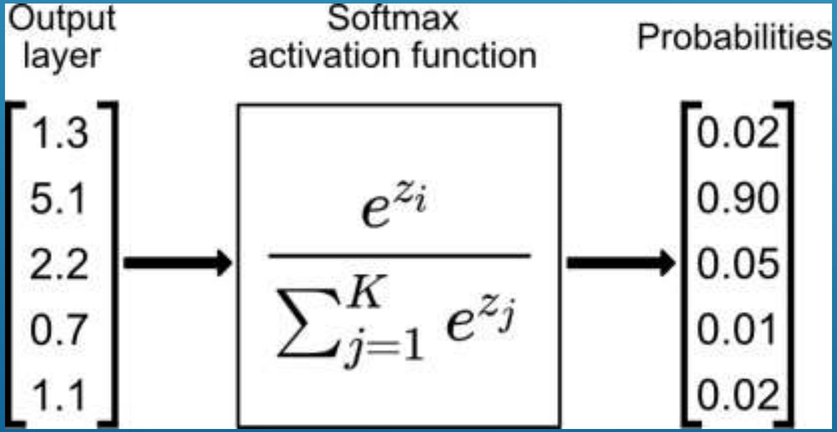


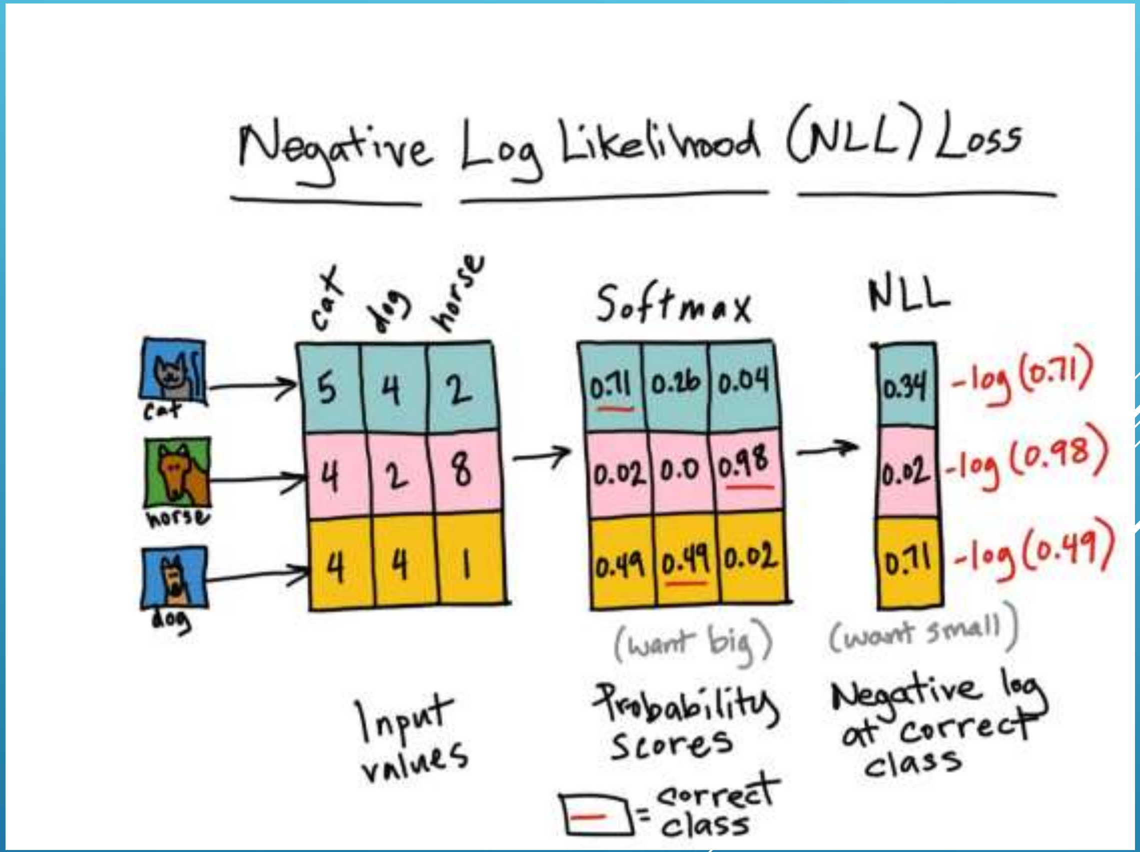
NEGATIVE LOG LIKELIHOOD LOSS FUNCTION, SOFTMAX & ADAM OPTIMIZER

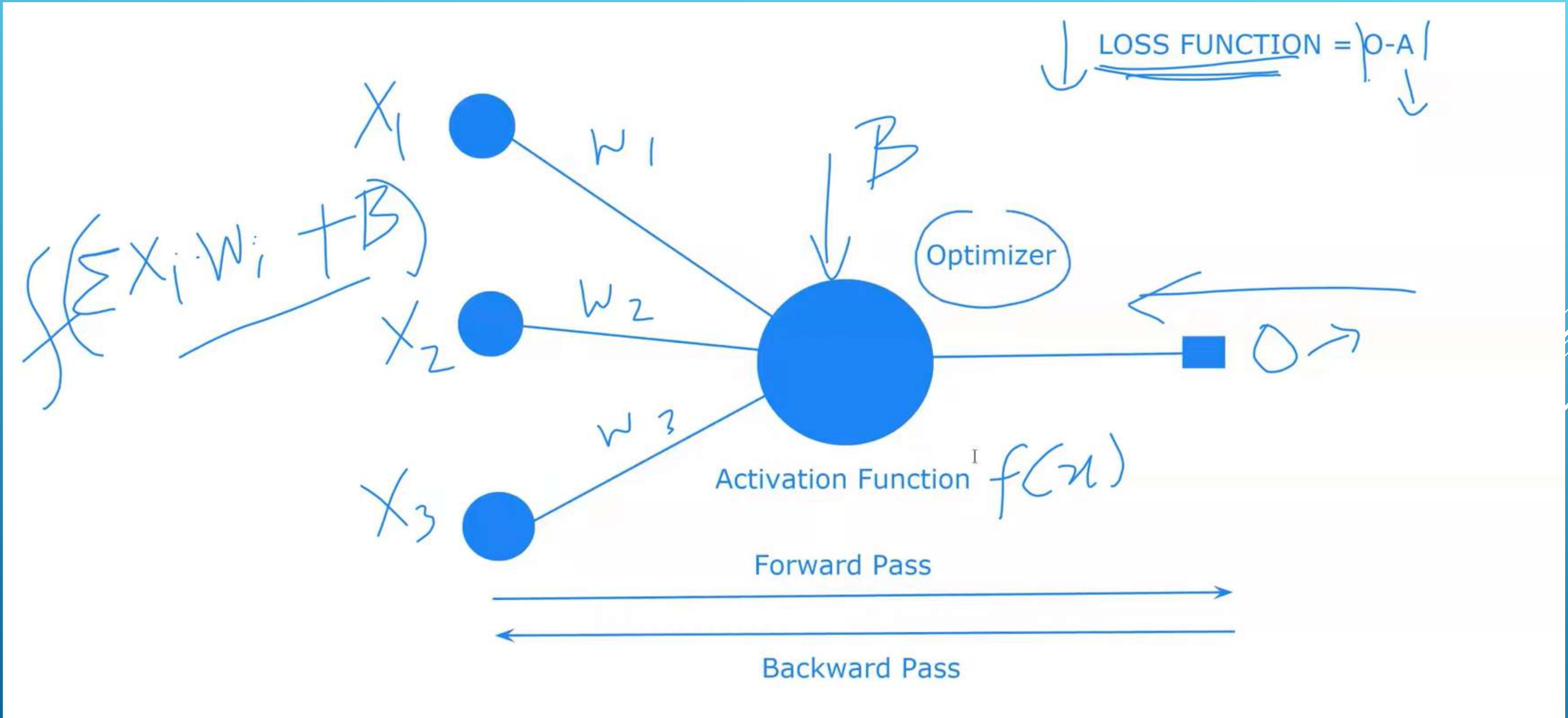
- Loss functions define what a good prediction is and isn't a way to measure how well the model is performing.

- Optimizers are algorithms or methods used to change the attributes of the neural network such as weights and learning rate (0.003) to reduce the losses.

- Adam is the best among the adaptive optimizers in most of the cases. Good with sparse data. a combination of RMSprop and Stochastic Gradient Descent with momentum. It uses the squared gradients to scale the learning rate like RMSprop and it takes advantage of momentum by using moving average of the gradient instead of gradient itself like SGD with momentum.






LOSS FUNCTION = |O-A|
X_1
W_1
B
$f(\sum X_i \cdot W_i + B)$
W_2
X_2
Optimizer
O
W_3
Activation Function
$f(x)$
X_3
Forward Pass
Backward Pass

Research Breif

- ❑ Compared many Augmentation libraries, and chosen the efficient features of it and built an custom augmentation 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

Research Breif

- ❑ Implemented the RESNET18 model

- ❑Ran for 100 epochs for each test case [Augmented Dataset]
& choosing the best trained model using the test accuracies

- ❑Ran the test cases for 1:2, 1:4, 1:8, 1:16, 1:20 imbalance

- ❑Ran the tests for 3 datasets

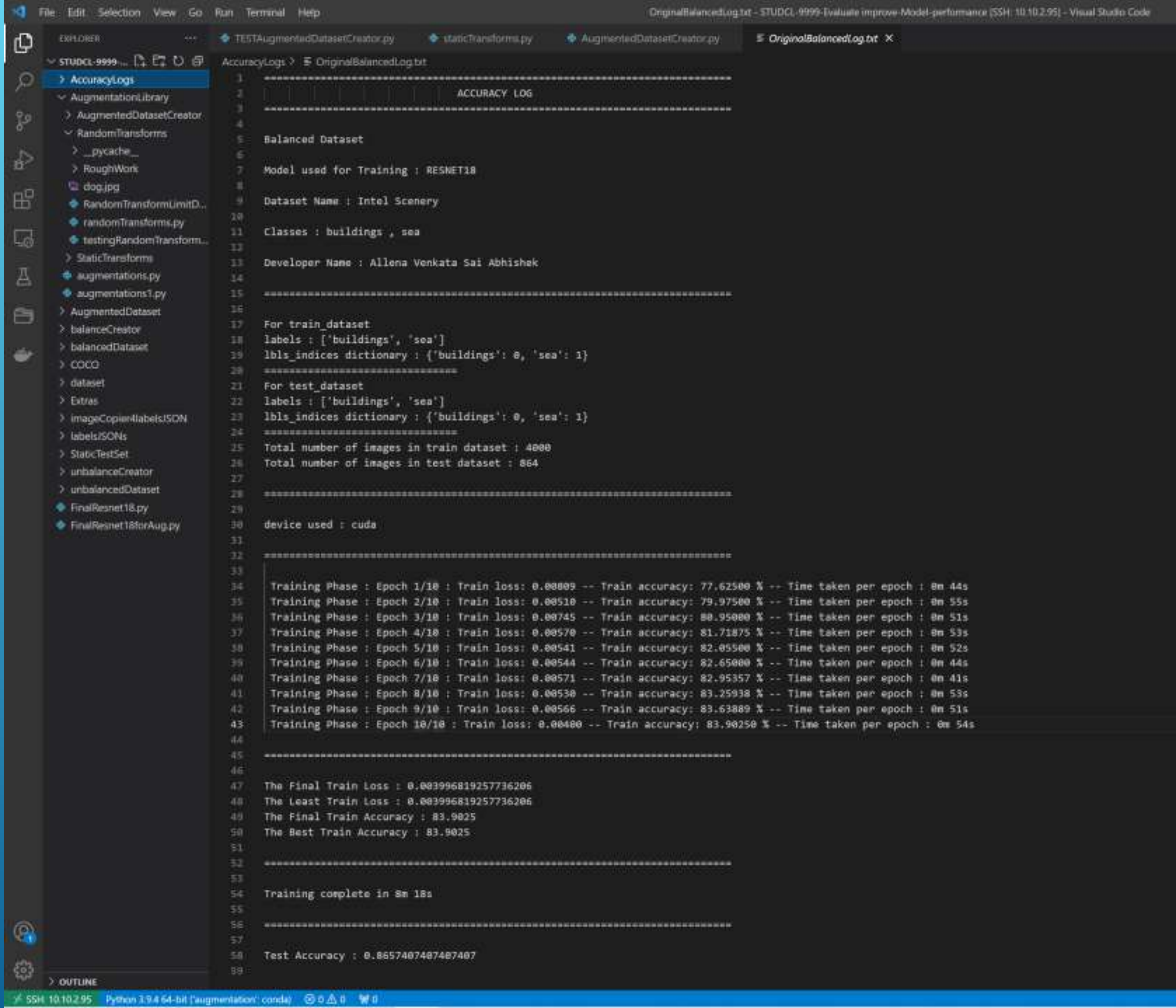
- ❑Analyzed & compared the results to compare the augmentations that improve the model accuracy

RESULTS?

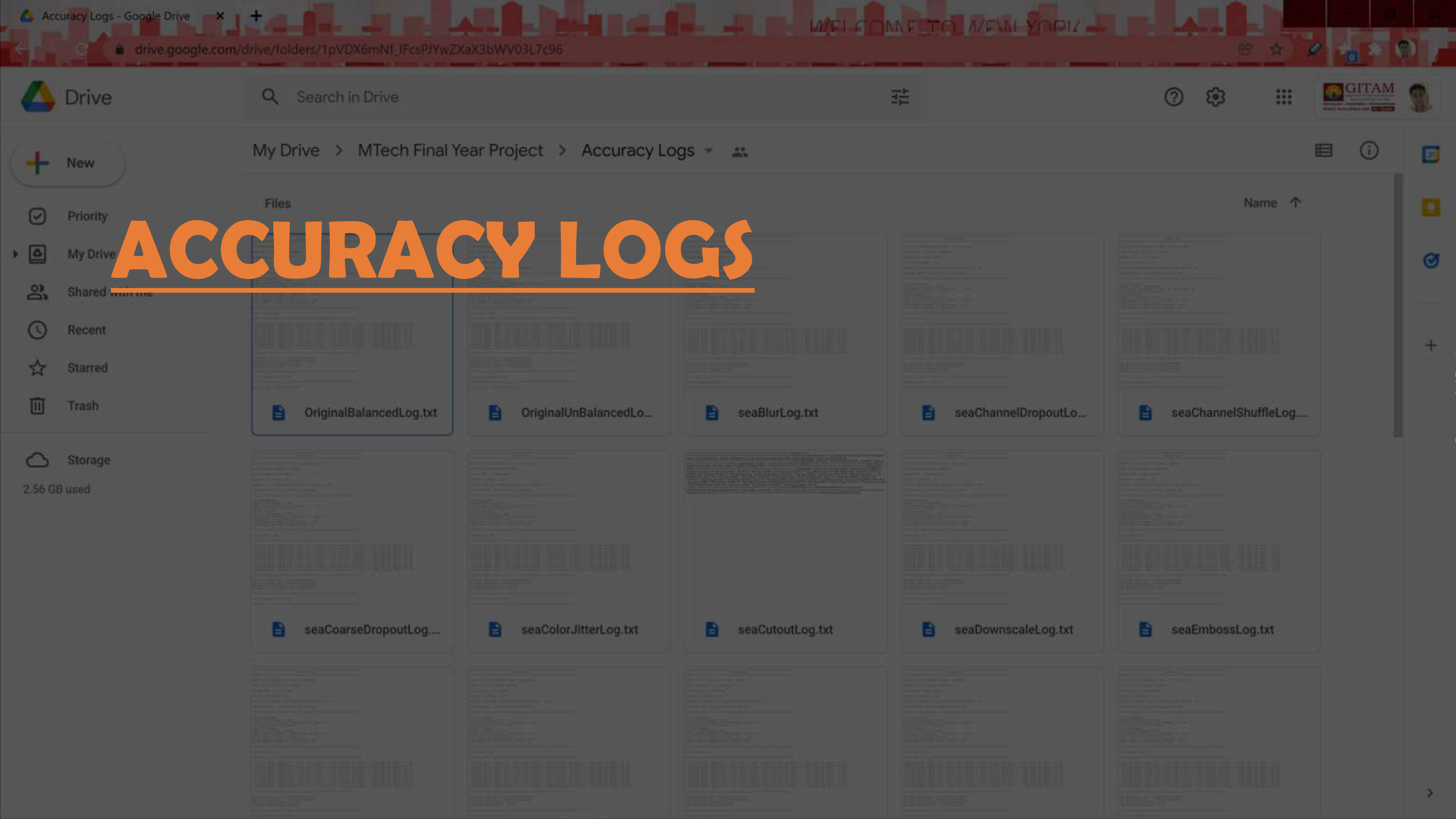
- ❑Finding accuracies on a fixed test set for all the test cases in which we are training using different test cases

- ❑Evaluating the model accuracies for each test case saved in accuracy log files

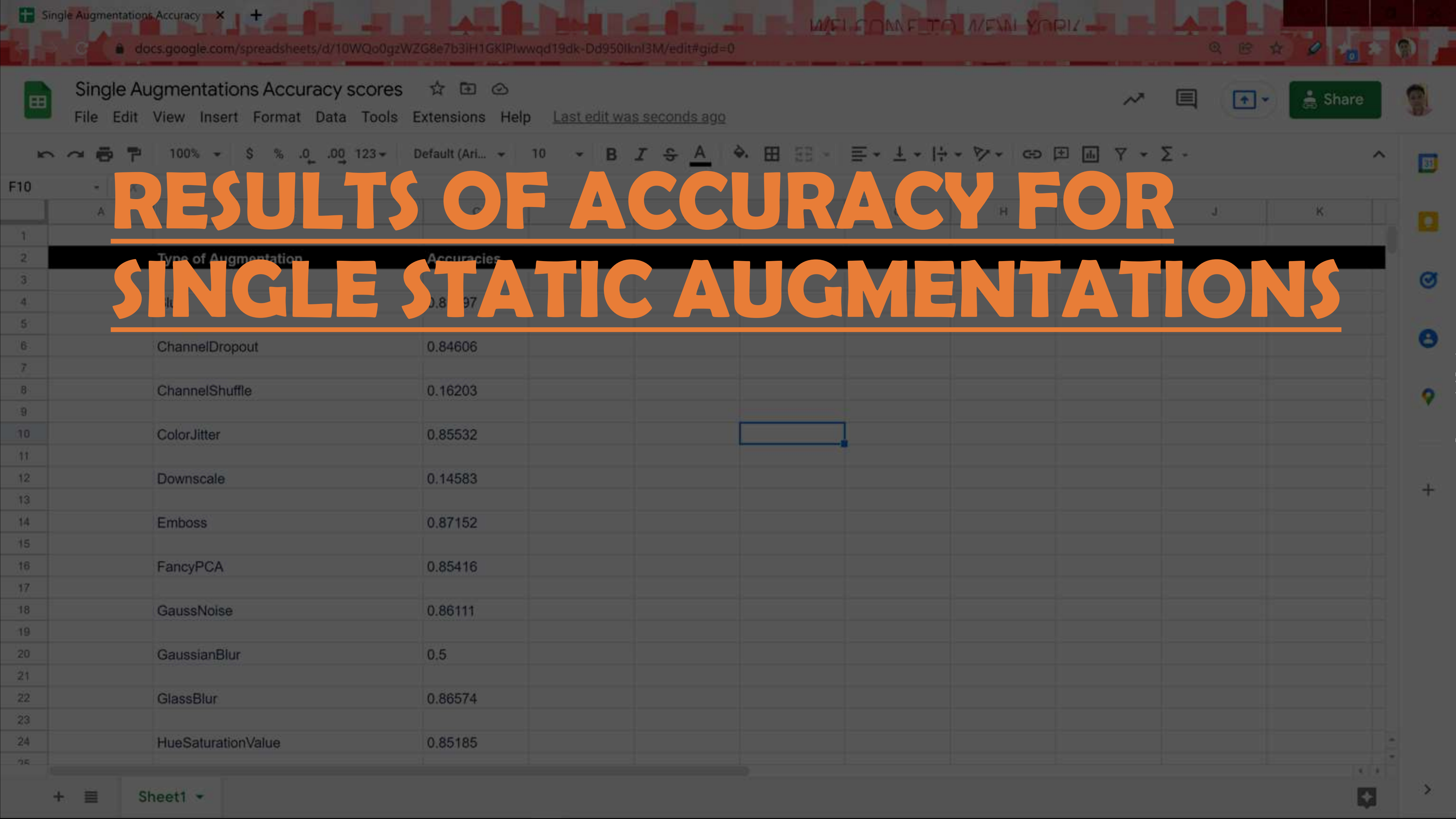
- ❑Finding out the best Augmentation techniques that gives the best accuracy for the Intel Scenery Dataset



ACCURACY LOGS



RESULTS OF ACCURACY FOR SINGLE STATIC AUGMENTATIONS



Intel Scene Classification dataset Single Augmentations Accuracy scores - Excel

Above Max Threshold	
Below Min Threshold	

2:1 (2000:1000)

BASE MAXIMUM ACCURACY	0.86574
BASE MINIMUM ACCURACY	0.84722

	Type of Augmentation	Accuracies
1	CoarseDropout	0.87615
2	Emboss	0.87152
3	Blur	0.86921
4	InvertImg	0.86805
5	GlassBlur	0.86574
6	GaussNoise	0.86111
7	RGBShift	0.86111
8	ISONoise	0.85995
9	HorizontalFlip	0.85995
10	CLAHE	0.85612
11	ColorJitter	0.85532
12	Solarize	0.85532
13	FancyPCA	0.85416
14	OpticalDistortion	0.85416
15	Posterize	0.853
16	HueSaturationValue	0.85185
17	ChannelShuffle	0.84837
18	MotionBlur	0.84722
19	JpegCompression	0.84722
20	ChannelDropout	0.84606
21	MultiplicativeNoise	0.84606
22	GridDistortion	0.84259
23	Sharpen	0.83912
24	VerticalFlip	0.83912
25	Superpixels	0.8368
26	GaussianBlur	0.83101
27	MedianBlur	0.8287
28	Cutout	0.8287
29	Transpose	0.82407
30	Downscale	0.82175

4:1 (2000:500)

BASE MAXIMUM ACCURACY	0.86574
BASE MINIMUM ACCURACY	0.79745

	Type of Augmentation	Accuracies
1	JpegCompression	0.86689
2	RGBShift	0.86574
3	HorizontalFlip	0.86458
4	MultiplicativeNoise	0.86342
5	GaussNoise	0.86111
6	GridDistortion	0.85069
7	InvertImg	0.84953
8	HueSaturationValue	0.84837
9	OpticalDistortion	0.84606
10	Blur	0.8449
11	CoarseDropout	0.84375
12	CLAHE	0.83912
13	VerticalFlip	0.83912
14	FancyPCA	0.83333
15	MotionBlur	0.83217
16	ToSepia	0.83217
17	Emboss	0.82986
18	ColorJitter	0.82754
19	Downscale	0.82407
20	Transpose	0.82407
21	MedianBlur	0.82291
22	Posterize	0.82291
23	ISONoise	0.81944
24	GlassBlur	0.81365
25	Superpixels	0.80324
26	ToGray	0.80208
27	Cutout	0.79976
28	GaussianBlur	0.7905
29	ChannelDropout	0.77083
30	GridDropout	0.77083

8:1 (2000:250)

BASE MAXIMUM ACCURACY	0.86574
BASE MINIMUM ACCURACY	0.65046

	Type of Augmentation	Accuracies
1	ColorJitter	0.86226
2	JpegCompression	0.86226
3	RGBShift	0.86111
4	GaussNoise	0.85995
5	Superpixels	0.85879
6	InvertImg	0.85763
7	Downscale	0.85185
8	FancyPCA	0.85069
9	OpticalDistortion	0.84837
10	VerticalFlip	0.84722
11	MultiplicativeNoise	0.8449
12	ISONoise	0.84259
13	Transpose	0.84259
14	Emboss	0.84027
15	Posterize	0.83912
16	HorizontalFlip	0.8368
17	CoarseDropout	0.8368
18	MotionBlur	0.83101
19	HueSaturationValue	0.82986
20	MedianBlur	0.82523
21	ChannelShuffle	0.82407
22	GridDistortion	0.82407
23	ToGray	0.81944
24	Blur	0.79513
25	Cutout	0.78819
26	GaussianBlur	0.78703
27	GlassBlur	0.78472
28	CLAHE	0.7662
29	ToSepia	0.74537
30	ChannelDropout	0.73958

16:1 (2000:125)

BASE MAXIMUM ACCURACY	0.86574
BASE MINIMUM ACCURACY	0.59143

	Type of Augmentation	Accuracies
1	InvertImg	0.87731
2	VerticalFlip	0.86921
3	HorizontalFlip	0.86574
4	FancyPCA	0.86458
5	GaussNoise	0.86458
6	RGBShift	0.85648
7	GridDistortion	0.84375
8	HueSaturationValue	0.84143
9	MultiplicativeNoise	0.84143
10	MotionBlur	0.84027
11	Posterize	0.83796
12	Transpose	0.83796
13	Downscale	0.8368
14	Emboss	0.83217
15	JpegCompression	0.83217
16	CoarseDropout	0.82754
17	ChannelShuffle	0.81597
18	ISONoise	0.81597
19	OpticalDistortion	0.80787
20	Superpixels	0.80671
21	ColorJitter	0.80092
22	MedianBlur	0.79513
23	Cutout	0.79513
24	ToGray	0.76504
25	GaussianBlur	0.74074
26	Blur	0.728
27	GlassBlur	0.68634
28	ToSepia	0.66898
29	ChannelDropout	0.61921
30	Solarize	0.60185

20:1 (2000:100)

BASE MAXIMUM ACCURACY	0.86574
BASE MINIMUM ACCURACY	0.58217

	Type of Augmentation	Accuracies
1	FancyPCA	0.87731
2	MultiplicativeNoise	0.86805
3	HueSaturationValue	0.86458
4	RGBShift	0.86342
5	VerticalFlip	0.86226
6	HorizontalFlip	0.85879
7	ColorJitter	0.85648
8	Posterize	0.85532
9	JpegCompression	0.85069
10	Downscale	0.84143
11	Transpose	0.84027
12	ISONoise	0.83912
13	ChannelShuffle	0.82986
14	Emboss	0.82175
15	GaussNoise	0.81828
16	Superpixels	0.81828
17	InvertImg	0.81134
18	GridDistortion	0.81134
19	Cutout	0.79745
20	MotionBlur	0.79398
21	OpticalDistortion	0.78125
22	CoarseDropout	0.7662
23	ToGray	0.76388
24	GlassBlur	0.75231
25	MedianBlur	0.73726
26	Blur	0.7118
27	ToSepia	0.71064
28	GaussianBlur	0.70138
29	ChannelDropout	0.66319
30	CLAHE	0.63078

INTEL SCENE CLASSIFICATION DATA | FINAL METRICS | 21 | 41 | 81 | 161 | 201 | Sheet8

Chest Xray Pneumonia Single Augmentations Accuracy scores

Above Max Threshold	
Below Min Threshold	

Balanced the 2:1

BASE MAXIMUM ACCURACY	0.85042
BASE MINIMUM ACCURACY	0.83974

	Type of Augmentation	Accuracies
1	ToSepia	0.85683
2	VerticalFlip	0.8547
3	MotionBlur	0.85256
4	GaussNoise	0.84401
5	ChannelShuffle	0.84188
6	Posterize	0.84188
7	OpticalDistortion	0.84188
8	GridDropout	0.84188
9	ColorJitter	0.83974
10	ToGray	0.83974
11	Transpose	0.83974
12	Emboss	0.83333
13	JpegCompression	0.83119
14	CLAHE	0.82905
15	Downscale	0.82905
16	ISONoise	0.82905
17	Superpixels	0.82905
18	Blur	0.82692
19	HueSaturationValue	0.82692
20	Cutout	0.82692
21	MedianBlur	0.82478
22	GridDistortion	0.82478
23	FancyPCA	0.82264
24	InvertImg	0.82264
25	Sharpen	0.82264
26	MultiplicativeNoise	0.82051
27	CoarseDropout	0.81837
28	GlassBlur	0.81623
29	RGBShift	0.81623
30	HorizontalFlip	0.81623
31	ChannelDropout	0.80982
32	Solarize	0.80128
33	GaussianBlur	0.594

Balanced the 4:1

BASE MAXIMUM ACCURACY	0.84188
BASE MINIMUM ACCURACY	0.83974

	Type of Augmentation	Accuracies
1	InvertImg	0.85897
2	Sharpen	0.85897
3	Transpose	0.85042
4	GaussianBlur	0.8482
5	GridDistortion	0.84615
6	HueSaturationValue	0.8461
7	ChannelDropout	0.846
8	CLAHE	0.846
9	RGBShift	0.84401
10	Solarize	0.84188
11	Superpixels	0.84188
12	ChannelShuffle	0.841
13	MedianBlur	0.8397
14	Posterize	0.83547
15	VerticalFlip	0.83547
16	JpegCompression	0.8354
17	CoarseDropout	0.8347
18	ToGray	0.83333
19	ToSepia	0.83333
20	Cutout	0.83333
21	MotionBlur	0.83119
22	OpticalDistortion	0.83119
23	ISONoise	0.82905
24	GridDropout	0.82905
25	Blur	0.829
26	FancyPCA	0.8269
27	GlassBlur	0.8226
28	ColorJitter	0.822
29	HorizontalFlip	0.82051
30	Downscale	0.8205
31	MultiplicativeNoise	0.81623
32	Emboss	0.8162
33	GaussNoise	0.8076

Balanced the 8:1

BASE MAXIMUM ACCURACY	0.83974
BASE MINIMUM ACCURACY	0.80555

	Type of Augmentation	Accuracies
1	VerticalFlip	0.84615
2	ColorJitter	0.84401
3	Transpose	0.83974
4	ChannelDropout	0.83547
5	Superpixels	0.83547
6	Emboss	0.83333
7	GaussNoise	0.83333
8	Cutout	0.83333
9	GridDistortion	0.83119
10	GridDropout	0.83119
11	MultiplicativeNoise	0.82905
12	HorizontalFlip	0.82905
13	JpegCompression	0.82905
14	MedianBlur	0.82692
15	FancyPCA	0.82478
16	Sharpen	0.82478
17	ChannelShuffle	0.82264
18	ISONoise	0.82264
19	InvertImg	0.82264
20	OpticalDistortion	0.82264
21	Blur	0.82051
22	Posterize	0.82051
23	Downscale	0.81837
24	GaussianBlur	0.81837
25	MotionBlur	0.81837
26	ToSepia	0.81837
27	Solarize	0.81623
28	CoarseDropout	0.81623
29	RGBShift	0.8141
30	HueSaturationValue	0.81196
31	ToGray	0.80982
32	CLAHE	0.80769
33	GlassBlur	0.79273

Balanced the 16:1

BASE MAXIMUM ACCURACY	0.83974
BASE MINIMUM ACCURACY	0.8

	Type of Augmentation	Accuracies
1	Emboss	0.84615
2	GridDistortion	0.83547
3	Transpose	0.83333
4	Downscale	0.83119
5	JpegCompression	0.83119
6	ToGray	0.82905
7	GaussNoise	0.82692
8	VerticalFlip	0.82692
9	ChannelShuffle	0.82478
10	FancyPCA	0.82478
11	OpticalDistortion	0.82478
12	CoarseDropout	0.82478
13	Cutout	0.82051
14	MultiplicativeNoise	0.81837
15	ColorJitter	0.81623
16	MedianBlur	0.8141
17	MotionBlur	0.8141
18	Posterize	0.8141
19	HueSaturationValue	0.80982
20	ISONoise	0.80769
21	RGBShift	0.80769
22	Sharpen	0.80769
23	Superpixels	0.80769
24	HorizontalFlip	0.80769
25	InvertImg	0.80555
26	GaussianBlur	0.80341
27	Blur	0.79914
28	Solarize	0.79487
29	GridDropout	0.79273
30	ChannelDropout	0.79059
31	GlassBlur	0.79059
32	CLAHE	0.76068
33	ToSepia	0.74358

Balanced the 20:1

BASE MAXIMUM ACCURACY	0.83974
BASE MINIMUM ACCURACY	0.79059

	Type of Augmentation	Accuracies
1	GridDistortion	0.84401
2	VerticalFlip	0.83974
3	MotionBlur	0.83547
4	Sharpen	0.83333
5	ChannelDropout	0.83119
6	HueSaturationValue	0.83119
7	JpegCompression	0.82692
8	Downscale	0.82478
9	Transpose	0.82478
10	OpticalDistortion	0.82478
11	Emboss	0.82051
12	FancyPCA	0.82051
13	GaussNoise	0.82051
14	HorizontalFlip	0.82051
15	CoarseDropout	0.81837
16	GaussianBlur	0.81623
17	ToGray	0.81623
18	Blur	0.8141
19	InvertImg	0.8141
20	Superpixels	0.8141
21	MedianBlur	0.81196
22	GridDropout	0.81196
23	ColorJitter	0.80982
24	GlassBlur	0.80982
25	Posterize	0.80982
26	Cutout	0.80982
27	MultiplicativeNoise	0.80769
28	ChannelShuffle	0.80555
29	RGBShift	0.78418
30	ToSepia	0.78418
31	ISONoise	0.77136
32	Solarize	0.76282
33	CLAHE	0.75641

INTEL SCENE CLASSIFICATION DATA | FINAL METRICS | 21 | 41 | 81 | 161 | 201

CIFAR 10 Single Augmentations Accuracy scores - Excel

Above Max Threshold	
Below Min Threshold	

2:1 (2000:1000)

BASE MAXIMUM ACCURACY	0.86574
BASE MINIMUM ACCURACY	0.84722

	Type of Augmentation	Accuracies
1	CoarseDropout	0.87615
2	Emboss	0.87152
3	Blur	0.86921
4	InvertImg	0.86805
5	GlassBlur	0.86574
6	GaussNoise	0.86111
7	RGBShift	0.86111
8	ISONoise	0.85995
9	HorizontalFlip	0.85995
10	CLAHE	0.85612
11	ColorJitter	0.85532
12	Solarize	0.85532
13	FancyPCA	0.85416
14	OpticalDistortion	0.85416
15	Posterize	0.853
16	HueSaturationValue	0.85185
17	ChannelShuffle	0.84837
18	MotionBlur	0.84722
19	JpegCompression	0.84722
20	ChannelDropout	0.84606
21	MultiplicativeNoise	0.84606
22	GridDistortion	0.84259
23	Sharpen	0.83912
24	VerticalFlip	0.83912
25	Superpixels	0.8368
26	GaussianBlur	0.83101
27	MedianBlur	0.8287
28	Cutout	0.8287
29	Transpose	0.82407
30	Downscale	0.82175

4:1 (2000:500)

BASE MAXIMUM ACCURACY	0.86574
BASE MINIMUM ACCURACY	0.79745

	Type of Augmentation	Accuracies
1	JpegCompression	0.86689
2	RGBShift	0.86574
3	HorizontalFlip	0.86458
4	MultiplicativeNoise	0.86342
5	GaussNoise	0.86111
6	GridDistortion	0.85069
7	InvertImg	0.84953
8	HueSaturationValue	0.84837
9	OpticalDistortion	0.84606
10	Blur	0.8449
11	CoarseDropout	0.84375
12	CLAHE	0.83912
13	VerticalFlip	0.83912
14	FancyPCA	0.83333
15	MotionBlur	0.83217
16	ToSepia	0.83217
17	Emboss	0.82986
18	ColorJitter	0.82754
19	Downscale	0.82407
20	Transpose	0.82407
21	MedianBlur	0.82291
22	Posterize	0.82291
23	ISONoise	0.81944
24	GlassBlur	0.81365
25	Superpixels	0.80324
26	ToGray	0.80208
27	Cutout	0.79976
28	GaussianBlur	0.7905
29	ChannelDropout	0.77083
30	GridDropout	0.77083

8:1 (2000:250)

BASE MAXIMUM ACCURACY	0.86574
BASE MINIMUM ACCURACY	0.65046

	Type of Augmentation	Accuracies
1	ColorJitter	0.86226
2	JpegCompression	0.86226
3	RGBShift	0.86111
4	GaussNoise	0.85995
5	Superpixels	0.85879
6	InvertImg	0.85763
7	Downscale	0.85185
8	FancyPCA	0.85069
9	OpticalDistortion	0.84837
10	VerticalFlip	0.84722
11	MultiplicativeNoise	0.8449
12	ISONoise	0.84259
13	Transpose	0.84259
14	Emboss	0.84027
15	Posterize	0.83912
16	HorizontalFlip	0.8368
17	CoarseDropout	0.8368
18	MotionBlur	0.83101
19	HueSaturationValue	0.82986
20	MedianBlur	0.82523
21	ChannelShuffle	0.82407
22	GridDistortion	0.82407
23	ToGray	0.81944
24	Blur	0.79513
25	Cutout	0.78819
26	GaussianBlur	0.78703
27	GlassBlur	0.78472
28	CLAHE	0.7662
29	ToSepia	0.74537
30	ChannelDropout	0.73958

16:1 (2000:125)

BASE MAXIMUM ACCURACY	0.86574
BASE MINIMUM ACCURACY	0.59143

	Type of Augmentation	Accuracies
1	InvertImg	0.87731
2	VerticalFlip	0.86921
3	HorizontalFlip	0.86574
4	FancyPCA	0.86458
5	GaussNoise	0.86458
6	RGBShift	0.85648
7	GridDistortion	0.84375
8	HueSaturationValue	0.84143
9	MultiplicativeNoise	0.84143
10	MotionBlur	0.84027
11	Posterize	0.83796
12	Transpose	0.83796
13	Downscale	0.8368
14	Emboss	0.83217
15	JpegCompression	0.83217
16	CoarseDropout	0.82754
17	ChannelShuffle	0.81597
18	ISONoise	0.81597
19	OpticalDistortion	0.80787
20	Superpixels	0.80671
21	ColorJitter	0.80092
22	MedianBlur	0.79513
23	Cutout	0.79513
24	ToGray	0.76504
25	GaussianBlur	0.74074
26	Blur	0.728
27	GlassBlur	0.68634
28	ToSepia	0.66898
29	ChannelDropout	0.61921
30	Solarize	0.60185

20:1 (2000:100)

BASE MAXIMUM ACCURACY	0.86574
BASE MINIMUM ACCURACY	0.58217

	Type of Augmentation	Accuracies
1	FancyPCA	0.87731
2	MultiplicativeNoise	0.86805
3	HueSaturationValue	0.86458
4	RGBShift	0.86342
5	VerticalFlip	0.86226
6	HorizontalFlip	0.85879
7	ColorJitter	0.85648
8	Posterize	0.85532
9	JpegCompression	0.85069
10	Downscale	0.84143
11	Transpose	0.84027
12	ISONoise	0.83912
13	ChannelShuffle	0.82986
14	Emboss	0.82175
15	GaussNoise	0.81828
16	Superpixels	0.81828
17	InvertImg	0.81134
18	GridDistortion	0.81134
19	Cutout	0.79745
20	MotionBlur	0.79398
21	OpticalDistortion	0.78125
22	CoarseDropout	0.7662
23	ToGray	0.76388
24	GlassBlur	0.75231
25	MedianBlur	0.73726
26	Blur	0.7118
27	ToSepia	0.71064
28	GaussianBlur	0.70138
29	ChannelDropout	0.66319
30	CLAHE	0.63078

INTEL SCENE CLASSIFICATION DATA | FINAL METRICS | 21 | 41 | 81 | 161 | 201

Conclusion

Effectiveness of augmentations techniques for the datasets

For Intel Scene Dataset -

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

For Chest Xray Pneumonia Dataset -

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

For CIFAR 10 Dataset -

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

FUTURE SCOPE

Augmentation libraries and techniques have improved a lot over the past decade with an exponential increase in data. There is a lot of research going on & few scope for future are to implement additional features such as getting the Random Transformations with the type and parameter values of it, Implementing the Random Augmentations in Augmented Dataset Creator, Getting the Accuracies for Random Augmented Datasets, implementing more cases an 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. Then getting the accuracies of each for evaluation. GANs can also be implemented to create the augmented images. Data scientists & ML Engineers can use the system to improve the model to learn more with less 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.

References

1. **A systematic study of the class imbalance problem in convolutional neural networks -** *Mateusz Buda, Atsuto Maki, Maciej A. Mazurowski* (https://arxiv.org/pdf/1710.05381v2.pdf)

2. **Deep Residual Learning for Image Recognition** *Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun* (https://arxiv.org/pdf/1512.03385v1.pdf)

3. **A survey on Image Data Augmentation for Deep Learning** *Connor Taghi M. Khoshgoftaar* (https://journalofbigdata.springeropen.com/track/pdf/10.1186/s40537-019-0197-0.pdf)

4. **Radial-Based Undersampling for Imbalanced Data Classification** *Michał Koziarski* (https://arxiv.org/pdf/1906.00452v2.pdf)

5. **Augmentor: An Image Augmentation Library for Machine Learning** *Marcus D. Bloice, Christof Stocker, Andreas Holzinger* (https://arxiv.org/pdf/1708.04680.pdf)

6. **Albumentations: fast and flexible image Augmentations** *Alexandr A. Kalinin, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Alexander Buslaev* (https://www.mdpi.com/2078-2489/11/2/125)

7. **MixUp augmentation for image classification** *Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, David Lopez-Paz* (https://arxiv.org/pdf/1710.09412.pdf)

8. **Learning Data Manipulation for Augmentation and Weighting** *Zhiting Hu, Bowen Tan, Ruslan Salakhutdinov, Tom Mitchell, Eric P. Xing* (https://arxiv.org/pdf/1910.12795v1.pdf)

9. **Robust High-Resolution Video Matting with Temporal Guidance** *Shanchuan Lin, Linjie Yang, Imran Saleemi, Soumyadip Sengupta* (https://arxiv.org/pdf/2108.11515.pdf)

10. **Robust Retinal Vessel Segmentation from a Data Augmentation Perspective** *Xu Sun, Huihui Fang, Yehui Yang, Dongwei Zhu, Lei Wang, Junwei Liu, Yanwu Xu* (https://arxiv.org/pdf/2007.15883.pdf)

11. **The Effects of Class Imbalance and Training Data Size on Classifer Learning** *Wanwan Zheng, Mingzhe Jin* (https://link.springer.com/article/10.1007/s42979-020-0074-0)

12. **ImageNet Classification with Deep Convolutional Neural Networks** *Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton* (https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf)

13. **Generative Adversarial Networks** *Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio* (https://arxiv.org/pdf/1406.2661v1.pdf)

14. **SMOTE: Synthetic Minority Over-sampling Technique** *N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer* (https://arxiv.org/pdf/1106.1813v1.pdf)

15. **ResNet-like Architecture with Low Hardware Requirements** *Elena Limonova, Daniil Alfonso, Dmitry Nikolaev, Vladimir V. Arlazarov* (https://arxiv.org/pdf/2009.07190v2.pdf)

SRS (Software Requirement Specification)

- Text Editor : VS Code
- OS: Windows/Linux/MacOS
- Storage Space : 6GB
- RAM : 8GB
- GPU Required : Advisable, for running the model faster (NVIDIA GPU)

Libraries used

- Numpy (For arrays)
- PIL (Pillow/Python Image Library)
- OpenCV
- Tensorflow (Implement deep learning algorithms, since it allows us to take advantage of GPUs for more efficient training)
- Keras (offer simple and consistent APIS & for developing and evaluating deep learning models) [*Being a high-level API on top of TensorFlow, we can say that Keras makes TensorFlow easy*.]
- Matplotlib
- Sys
- Torch (Version - `'1.10.0'`)
- Pytorch (For using inbuilt transforms)
- Pprint
- Os
- Random
- Pandas (If need to store data in CSV etc)
- JSON (storing the data generated in a JSON etc)
- Albumentation (Using transforms)
- Glob
- Shutil

THANK YOU