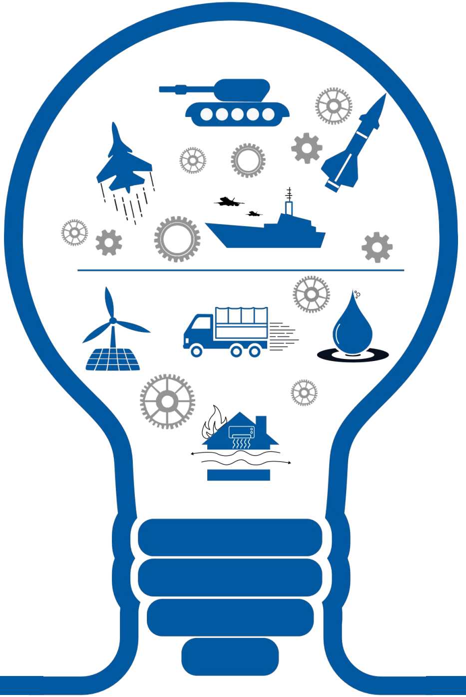
PP122

ZEUS NUMERIX

Applications & Capabilities in AI/ML

Target Recognition from Satellite Image Database

Irene Grace Karot Polson



Build-to-Specifications | Product Approval | Engineering Services | Software Development

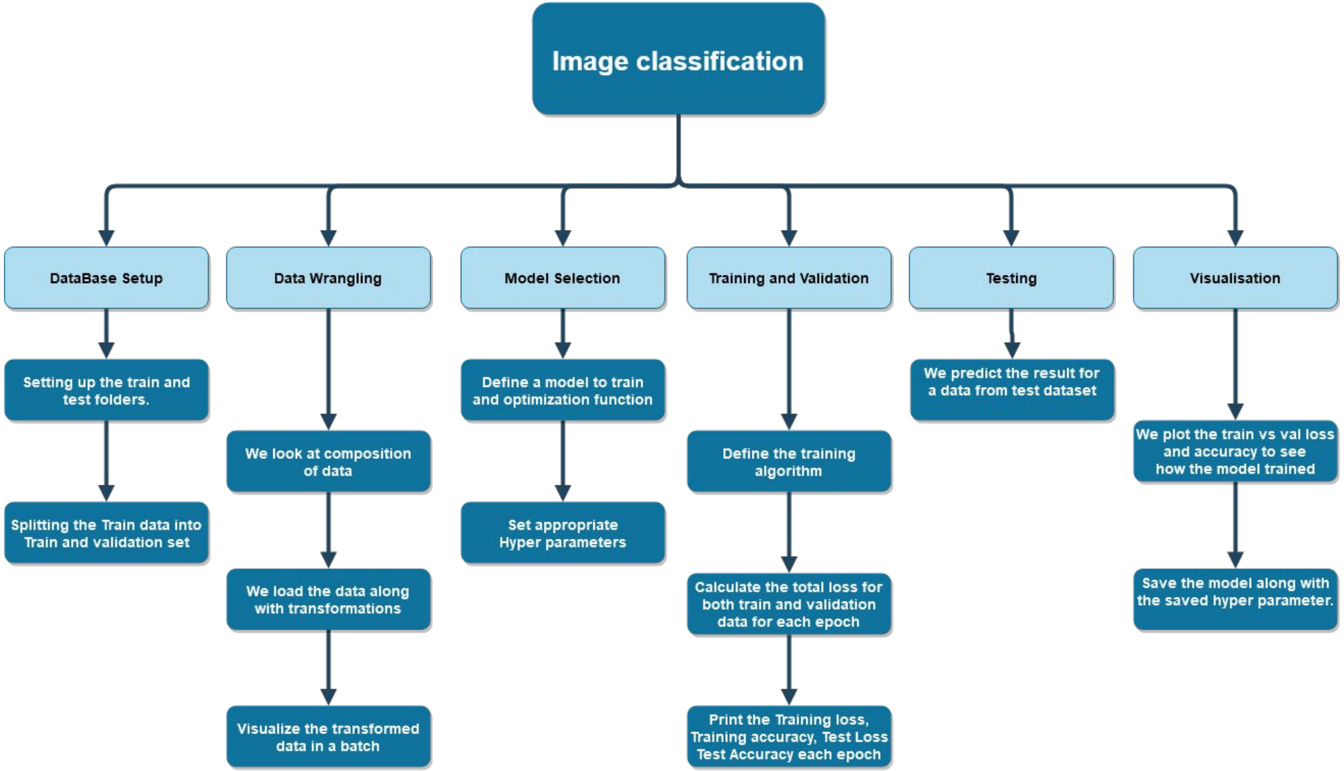
Contents

Content
Objective
Database Setup
Data Wrangling (Pre-processing)
Introduction to CNN Terminologies
Introduction to general ML Terminologies
Model Selection
Planes Satellite Images Dataset
Percentage White for Plane
Ship Images Database
Ship Satellite Images Dataset
Transfer Learning
Highway Vehicle Classification
SARS Image Classification
Conclusion

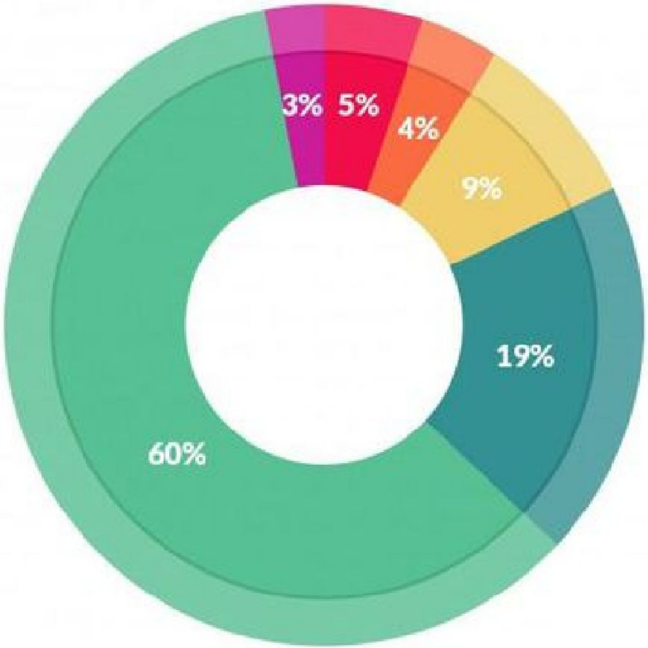
Objectives

- The main objective of the project is Target Classification of Satellite image data using Convolutional Neural Networks.
- We aim to identify and classify Satellite images of ships, but initially, I am implementing the CNN on a satellite image data to identify Planes.
- Develop a general program to which is independent of the Model and Optimiser used and can access the data, split it appropriate between the
- Perform this task on multiple models and compare results.
- Tune the Hyper Parameters to get better results in self defined models.

Basic Block Diagram of Image Classification

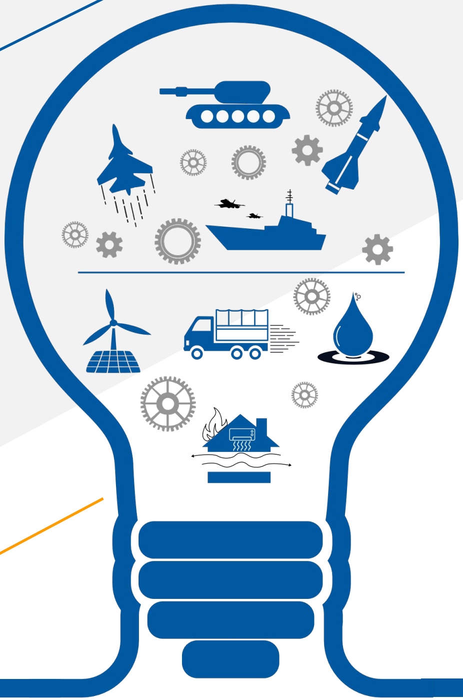


Forbes Survey on Data Scientists



- Building training datasets : 3%
- Cleaning and organising data : 60%
- Collecting datasets : 19%
- Mining data for patterns : 9%
- Refining algorithms : 4%
- Others : 5%

Source: https://www.forbes.com/sites/gilpress/2016/03/23/



Database Setup

Obtain database in required format

Database

- The satellite images of Planes from the **Kaggle**.
- They received the data from commercial imagery provider **Planet**.

- The aim of this dataset is to help address the difficult task of detecting the location of airplanes in satellite images. Automating this process can be applied to many issues including monitoring airports for activity and traffic patterns, and defense intelligence.

The Data include the following :

- **Label:** Valued 1 or 0, representing the "plane" class and "no-plane" class, respectively.
- **Scene id:** The unique identifier of the PlanetScope visual scene the image chip was extracted from. The scene id can be used with the Planet API to discover and download the entire scene.
- **Longitude_Latitude:** The longitude and latitude coordinates of the image center point, with values separated by a single underscore.

Database : Properties

- The dataset contains a directory with all the images, which we use.
- The dataset is also distributed as a JSON formatted text file *planesnet.json*. The loaded object contains data, label, scene_ids, and location lists.
- The pixel value data for each 20x20 RGB image is stored as a list of 1200 integers within the data list.

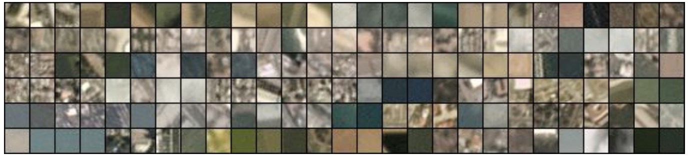
- The spatial resolution : 5m/pixel (of the RAPIDEYE satellite used for Open California imagery by Planet Imagery.)
- The average area covered in each image is around 1km^2 of the Earth

- The list values at index i in labels, scene_ids, and locations each correspond to the i-th image in the data list.

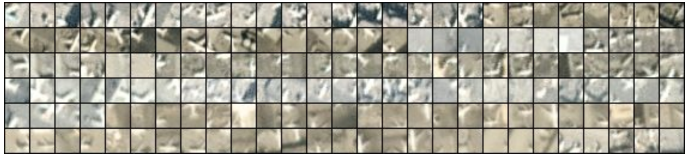
Database : Labels and Distribution

Planes

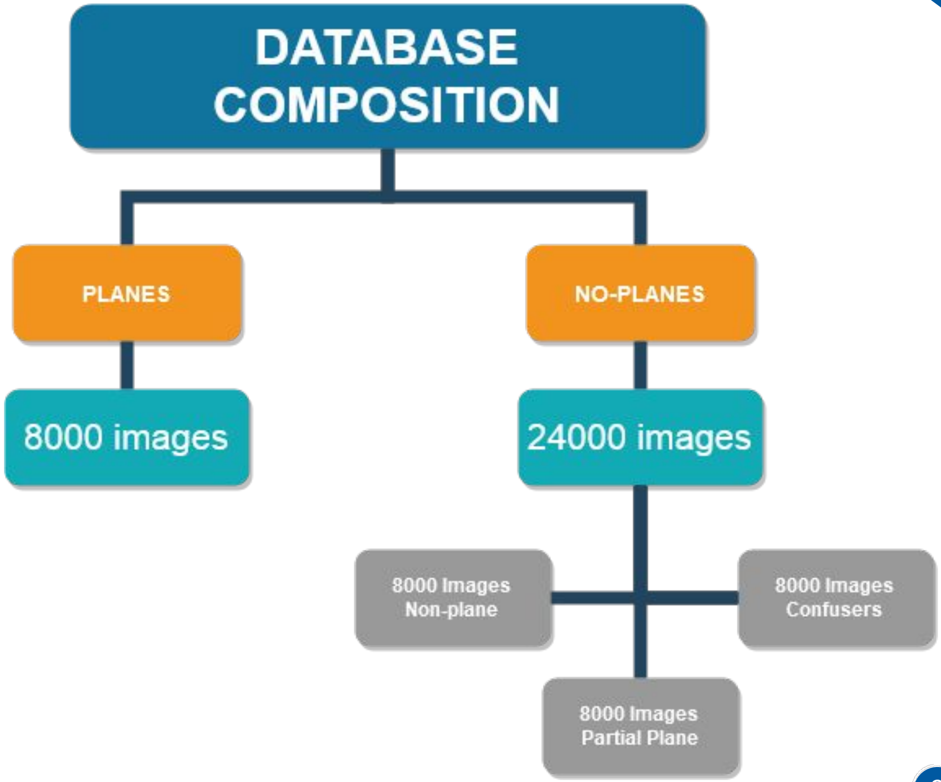
Non-Plane



Partial Planes



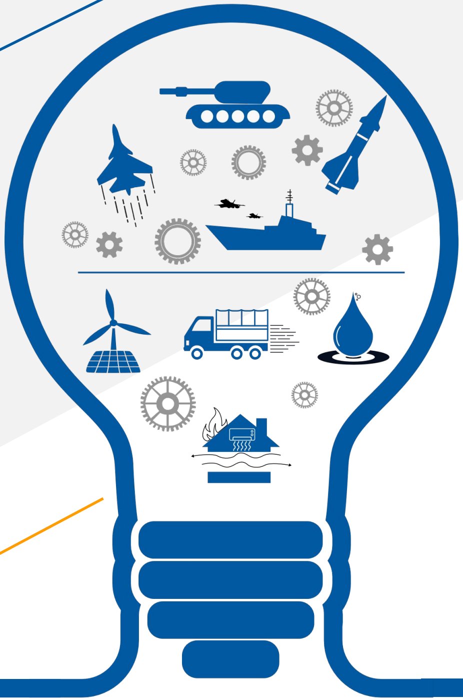
Confusing Images



Database : Issues

- The images are not separated into directories the way Pytorch requires for Data loading.

- The brightness of the images is high and the contrast is less. For the planes to be clearly visible, these factors need to be adjusted through transformations.



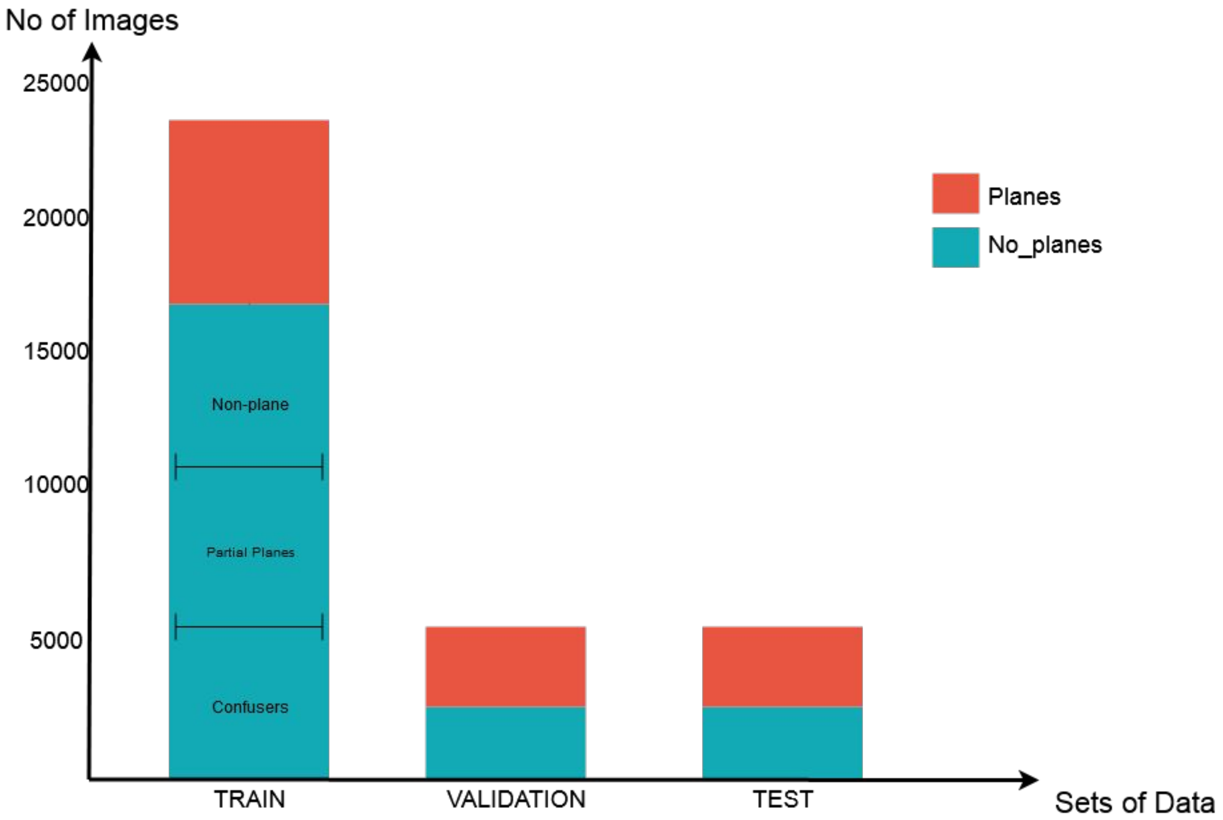
Data Wrangling

Obtain clean and model ready dataset

Data Wrangling

- Used the train_test_split function from the sklearn library to split my training data into train set, validation set and test set used in the model.
- We load the images batchwise(25 at a time) using the torch.nn function Dataloader which helps in iterating over the entire dataset applying the defined transformations to the images.
- The mean and std for the normalization are found by trial and error till the images of planes have sufficient contrast from the background.

Data Wrangling : Split of the Database

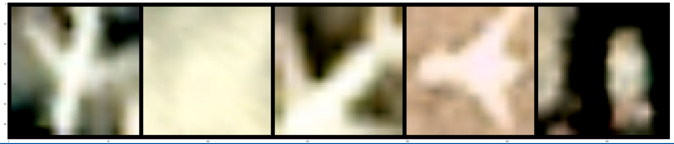


Data Wrangling : Load and visualise Data

- Transformations are defined that are applied to each image that is loaded.
  - Data augmentation : Horizontally flip random images to have variety in the training set, random crop.
  - Pixel Standardization: scale pixel values to have a zero mean and unit variance. The pixel values are centered around the mean with a unit standard deviation.

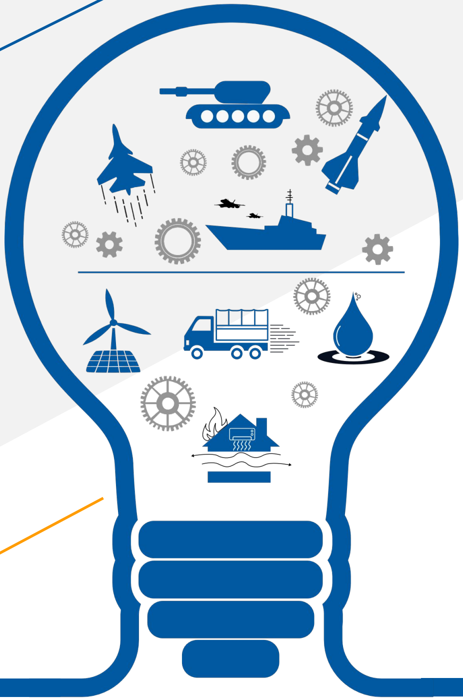
Before Transformations

After Transformations



$$z = \frac{x - \mu}{\sigma}$$

$x = value\ to\ be\ transformed$
$\mu = mean\ value\ of\ data$
$\sigma = standard\ deviation$



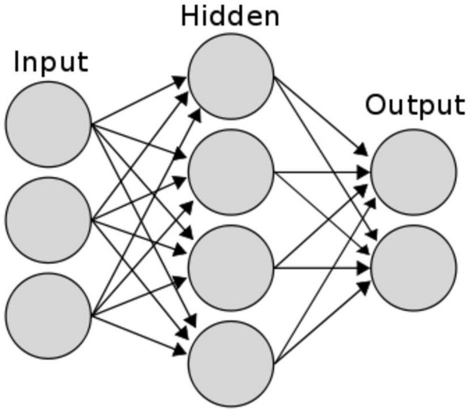
Introduction to CNN Terminologies

Understand the functions used to a general CNN model

Introduction : Neural Networks

- A neural network is a series of algorithms that tries to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.

- There are many ways to model an algorithm to work like the human brain. In levels of complexity :
  - Perceptron model
  - Multi-perceptron model
  - Development of Gradient descent, backpropagation
  - RNN, NLP, CNN

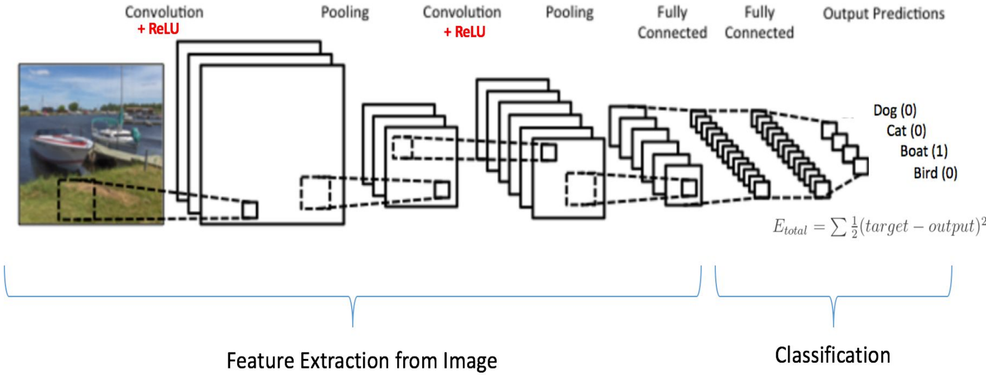


The model used in this project is CNN.

Introduction : CNN

- CNN - Convolutional Neural Network , as humans try to recognize objects in an image, we notice features like color, shape, and size that help you identify the objects.

- The main components in a CNN are :
  - the convolution
  - Activation function
  - pooling layer



where the model makes a note of the features in the image, and the fully connected (FC) layer, where classification takes place.

- Convolution is used in speech processing (1 dimension), image processing (2 dimensions), and video processing (3 dimensions).

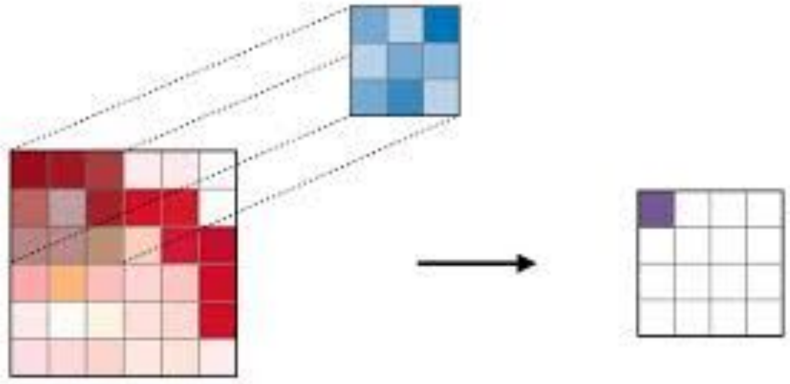
Introduction : Convolution Layer

The factors that affect the convolution layer output shape are :

- Kernel size (size of filter matrix)
- Input dimensions (no. of channels, RGB=3, greyscale=1)
- Padding
- Strides (no.of pixel by which the window moves).

$$W' = \frac{W - F + 2P}{S} + 1$$

$$W = WidthOfInputImage$$
$$F = KernelFilterWindowSize$$
$$P = Padding$$
$$S = Stride$$

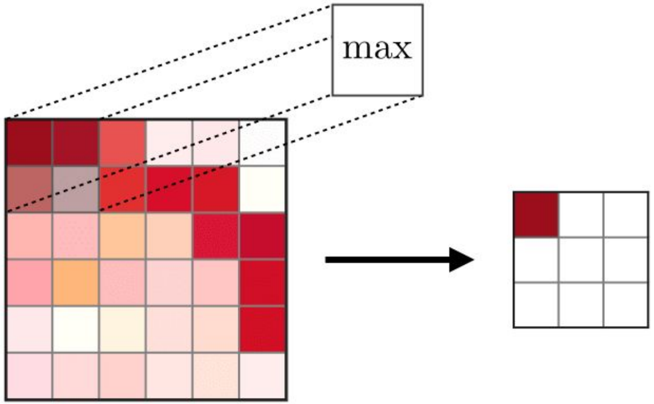


Source: https://stanford.edu/~shervine/teaching/cs-230/

Introduction : Pooling layer

- A drawback of a convolution feature map is that it records the exact position of features. Even the smallest development in the feature map will produce different results.

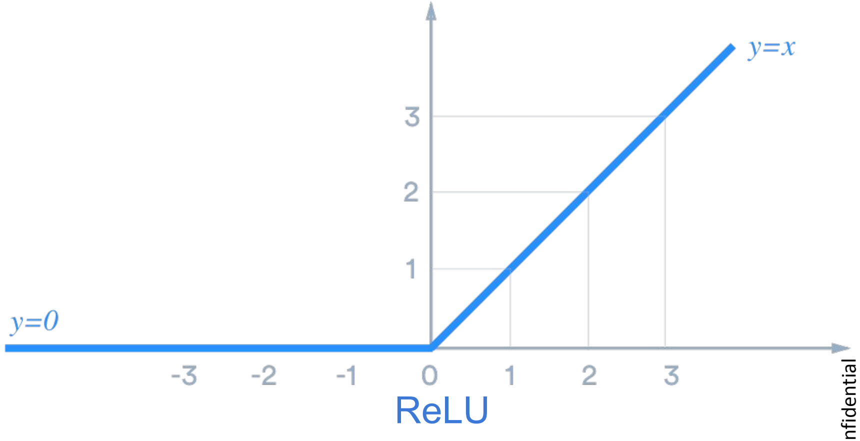
- Pooling layer will be a lower version of the image with important features intact.



Source: https://stanford.edu/~shervine/teaching/cs-230/

Introduction : Activation Function

- The activation functions help the network use the important information and suppress the irrelevant data points.
- Popular types of activation functions are :
  - Binary Step
  - Linear
  - Sigmoid
  - Tanh
  - ReLU
  - Leaky ReLU
  - Parameterised ReLU
  - Exponential Linear Unit
  - Swish
  - Softmax
- As a rule of thumb, we begin with using ReLU function and then move over to other activation functions in case ReLU doesn't provide with optimum results.



Source: https://medium.com/@danqing/a-practical-guide-to-relu

Introduction : Input and Output

In the convolution layer,

- Input : number of in channels.
  - For the first layer, it indicates the colour channels of the input image (RGB=3, B&W=1).
  - For the subsequent layers, it will be the previous number of out channels.
- Output : number of feature maps we want as out channels.

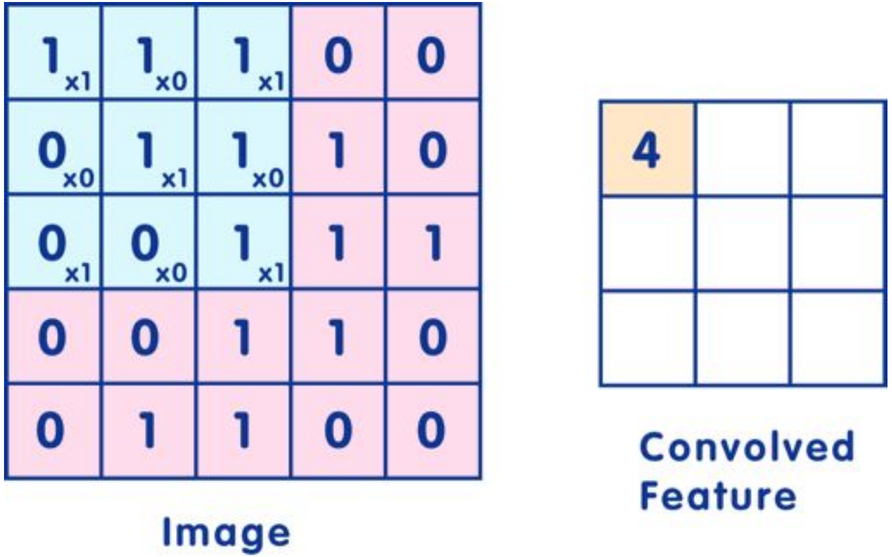
In the pooling layer,

- Input : It is the size of the image at each layer.
- Output : It will be the feature reduced to input/stride size.

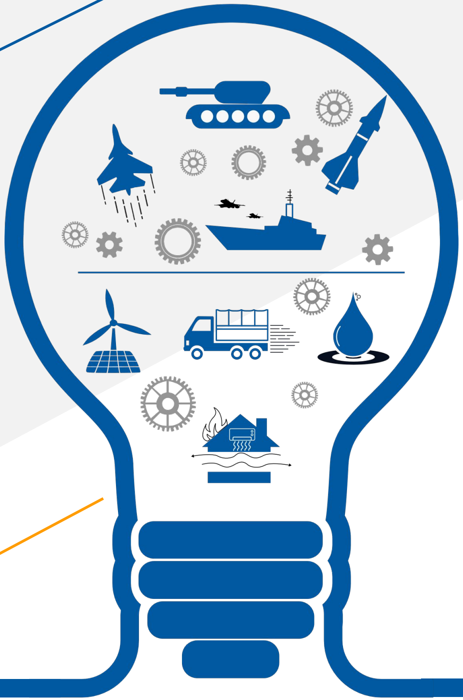
Introduction : Feature Extraction

- The different features of an image are identified and extracted using blurring, sharpening, embossing, edge detection, and more.

- In the case of CNN, we define the kernel sizes and let the model adjust its weights during optimisation such that that best feature kernels are formed and is able to identify the object clearly.



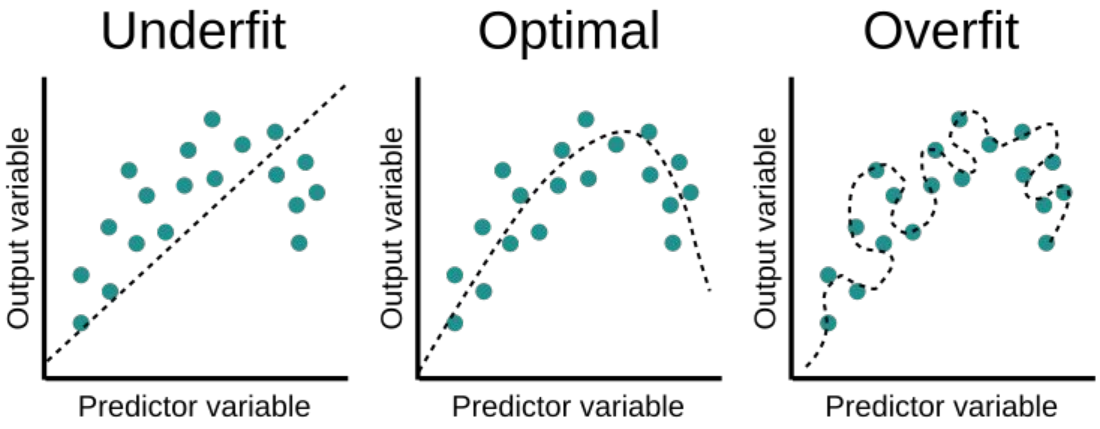
Source: https://austingwalters.com/convolutional-neural-networks-cnn-to-classify-sentences/cnn-filter/



Introduction to General ML Terminology

Introduction : Overfitting

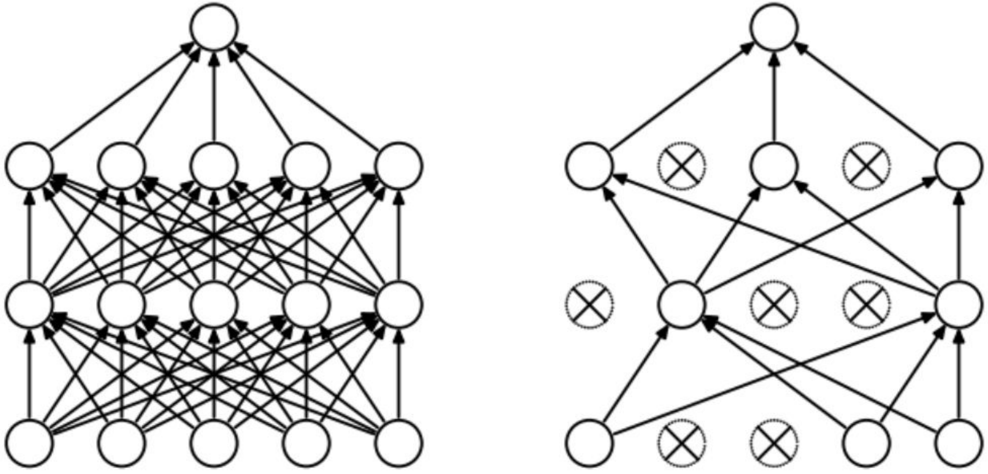
- The error on the training set is driven to a very small value, but when new data is presented to the network the error is large.
- The network has memorized the training examples, but it has not learned to generalize to new situations.



Source: https://www.educative.io/edpresso/overfitting-and-underfitting

Introduction : Dropout

- Dropout : Prevent Neural Networks from Overfitting by randomly setting the output for a given neuron to 0.

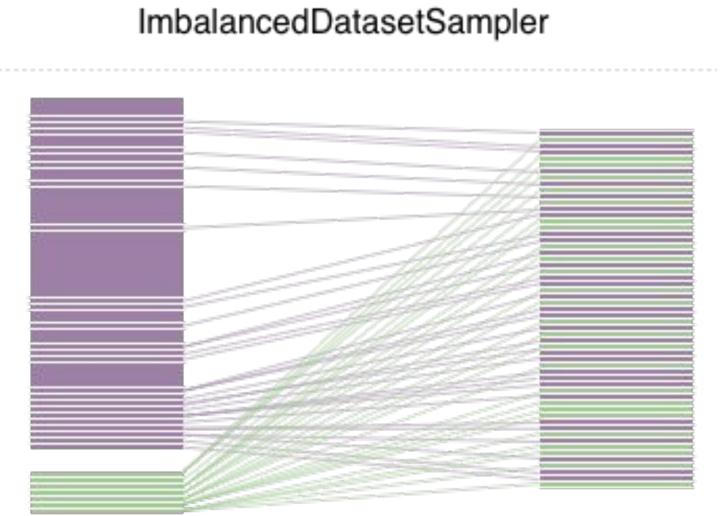


(a) Standard Neural Net (b) After applying dropout.

Source: https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning

Introduction: Subset Random Sampler

- This is a function to randomly sample images into uniform composition sets.
- We need to pass the labels to this function and it returns a set of label distribution.
- We later use this label distribution while loading the images into the training, validation and test.



Source:https://pythonawesome.com/a-pytorch-imbalanced-dataset-sampler-for-oversampling-low-frequent-class

Introduction : Loss

Loss functions map a set of parameter values for the network onto a scalar value that indicates how well those parameters fit the model.

There are different types of losses implemented in machine learning classifications :

Cross Entropy Loss

- if the prediction is 0, the first half goes away, and if the prediction is 1, the second half drops.

$$L = -(y\log(p) + (1-y)\log(1-p))$$

$$y = Actual Label$$
$$p = Predicted Probability$$

Introduction : Optimization

- Optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses.
- Common Optimisers :
    - Gradient Descent
    - Stochastic Gradient Descent
    - Mini-Batch Gradient Descent
    - Nesterov Accelerated Gradient
    - Adagrad
    - AdaDelta
    - Adam

If one wants to train the neural network in less time and more efficiently then Adam is the optimizer.

Loss function checks whether the model is moving in the correct direction and making progress, whereas optimization improves the model to deliver accurate results.

Introduction : Optimization

- The optimizers have some elements of the gradient descent.
- By changing the model parameters, like weights, and adding bias, the model can be optimized.
- The learning rate will decide how big the steps should be to change the parameters.

Steps in optimization :

1. Calculate what a small change in each weight would do to the loss function (selecting the direction to reach minima).
2. Adjust each weight based on its gradient (i.e., take a small step in the determined direction).
3. Keep doing steps 1 and 2 until the loss function gets as low as possible.

ADAM is a blend of RMSprop and stochastic gradient descent.

Introduction : Cosine Annealing LR

Adjust learning rate : Cosine Annealing LR

- Set the learning rate of each parameter group using a cosine annealing schedule.
- It changes the learning rate over the training iterations uniformly from the maximum to minimum defined learning rate.

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{T_{cur}}{T_{max}}\pi\right)\right)$$

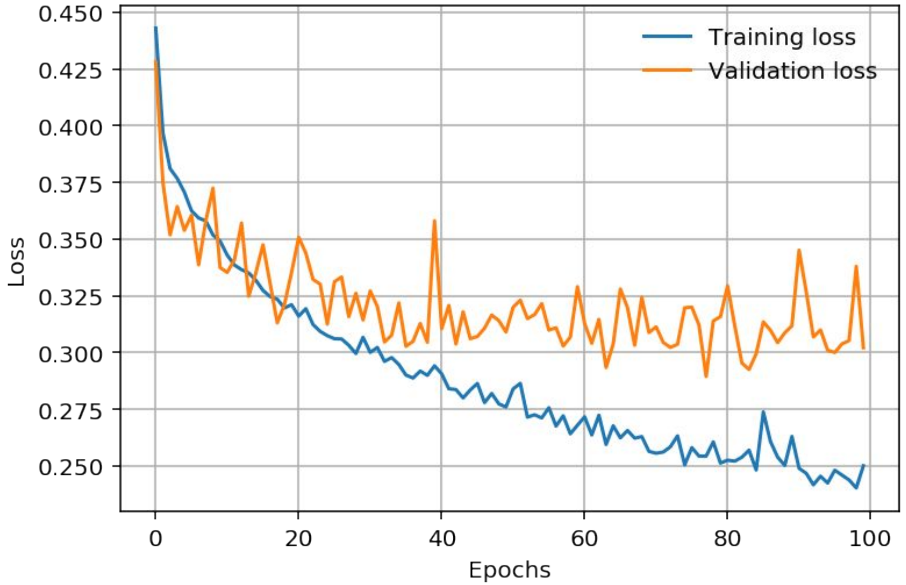
$$\eta = Learning\ Rate$$
$$T = Iteration\ in\ Epochs$$

Introduction : model.eval() Mode

This is when model one is trained without the validation in the model.eval() mode

The training and eval mode is decreasing the importance of the training set and increasing the importance of the validation set respectively.

When this is not applied, the model is very likely to overfit immediately.

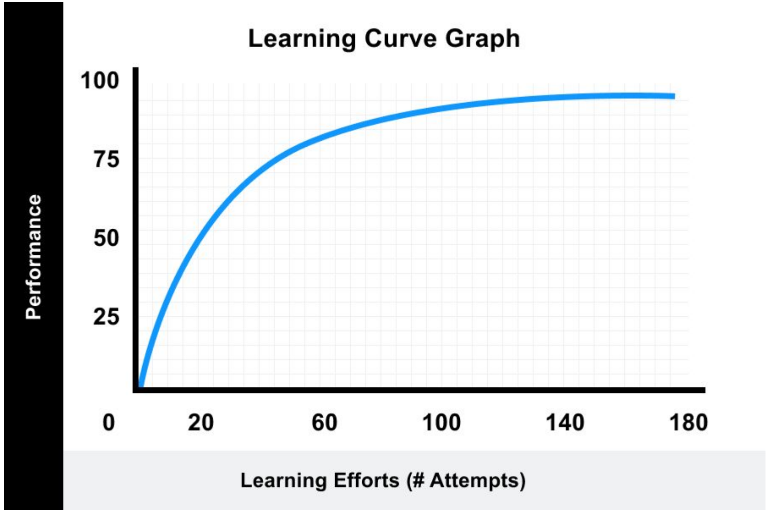


Learning Curves

Learning Curve: Line plot of learning (y-axis) over experience (x-axis).

- Train Learning Curve: Gives an idea of how well the model is learning.
- Validation Learning Curve: Gives an idea of how well the model is generalizing.

- Performance parameters : Error reduction, Loss reduction, Accuracy enhancement.
- Learning effort : Epochs, No of dataset samples.



Source: https://www.valamis.com/hub/learning-curve

Classification : Learning curves

In the case of classification predictive modeling problems, where the model may be optimized according to cross-entropy loss and model performance is evaluated using classification accuracy.

- Optimization Learning Curves: Learning curves calculated on the metric by which the parameters of the model are being optimized, e.g. loss.

- Performance Learning Curves: Learning curves calculated on the metric by which the model will be evaluated and selected, e.g. accuracy.

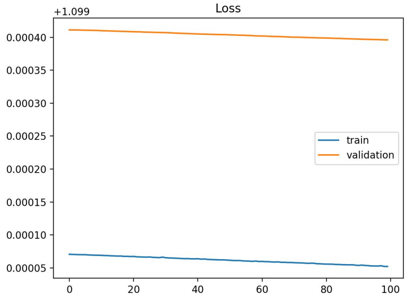
The problems that can be spotted by the learning curves are :

- Model Behavior :
  - Underfit
  - Overfit
- Unrepresentative Datasets

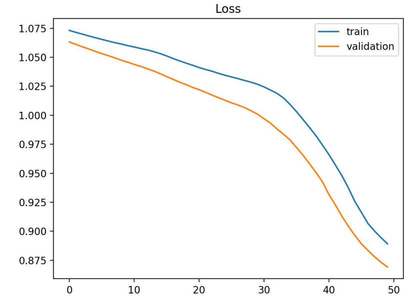
Learning curve : Underfit

The training loss may show a flat line or noisy values of relatively high loss, indicating that the model was unable to learn the training dataset at all.

- the model does not have a suitable capacity for the complexity of the dataset.



- An underfit model may also be identified by a training loss that is decreasing and continues to decrease at the end of the plot.

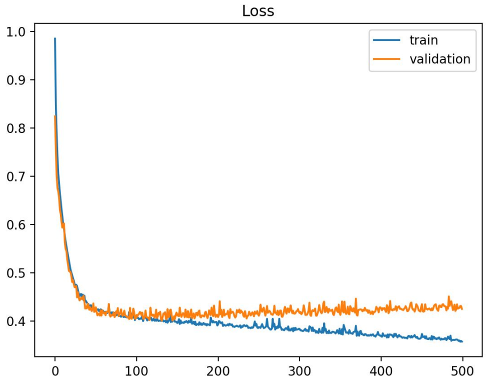


Source:https://machinelearningmastery.com

Learning curve : Overfit

A plot of learning curves shows overfitting if:

- The plot of training loss continues to decrease with experience.
- The plot of validation loss decreases to a point and begins increasing again.



Source:https://machinelearningmastery.com

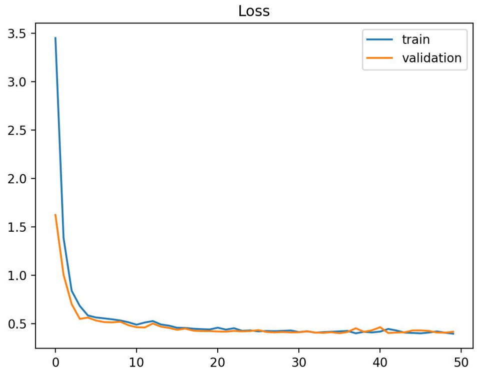
Learning curve : Goodfit

A good fit is identified by a training and validation loss that decreases to a point of stability with a minimal gap between the two final loss values.

The loss of the model will almost always be lower on the training dataset than the validation dataset. This means that we should expect some gap between the train and validation loss learning curves. This gap is referred to as the generalization gap.

A plot of learning curves shows a good fit if:

- The plot of training loss decreases to a point of stability.
- The plot of validation loss decreases to a point of stability and has a small gap with the training loss.

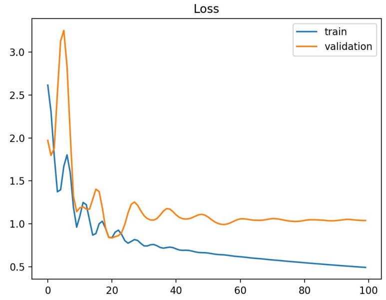


Source:https://machinelearningmastery.com

Learning curves : Unrepresentative datasets

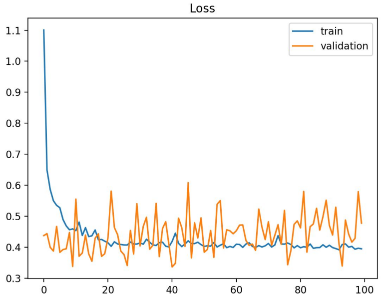
Unrepresentative Training Dataset

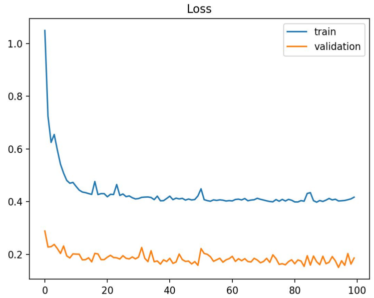
- Training dataset does not provide sufficient information to learn the problem, relative to the validation dataset used to evaluate it.
- This may occur if the training dataset has too few examples as compared to the validation dataset.



Unrepresentative Validation Dataset

- An unrepresentative validation dataset means that the validation dataset does not provide sufficient information to evaluate the ability of the model to generalize.
- This may occur if the validation dataset has too few examples as compared to the training dataset.



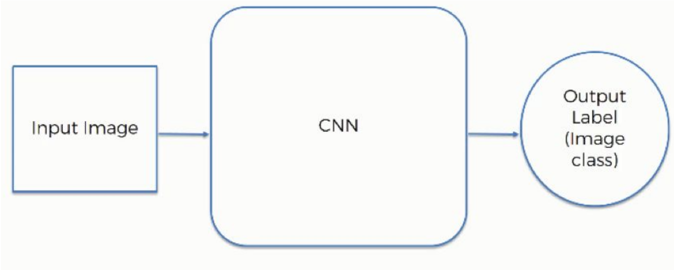


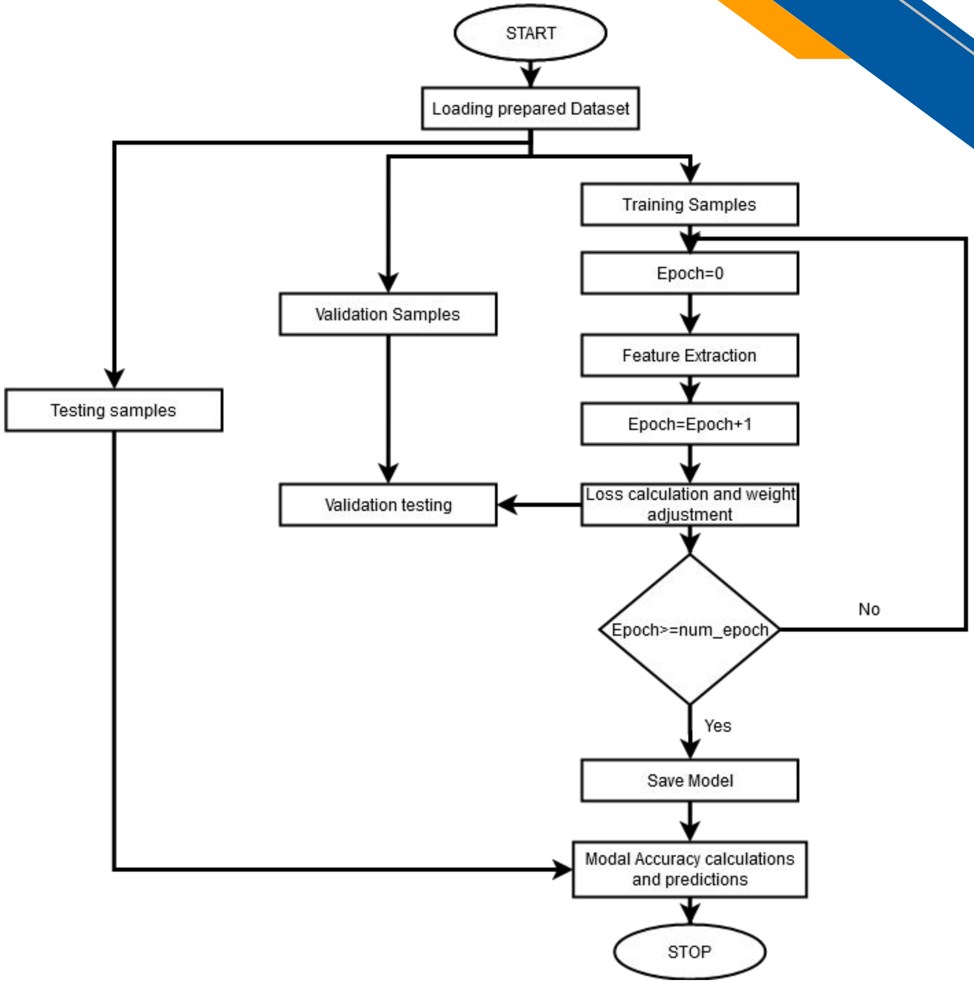
Source:https://machinelearningmastery.com

Model Selection

Obtaining Tuned hyperparameters

General CNN Flowchart





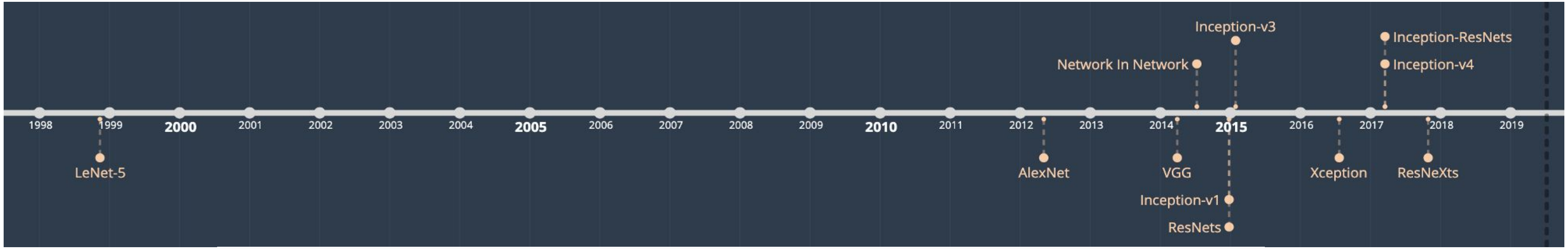
Model Selection

When making a CNN model, the parameters you can make differences in are :

- The number of Convolution Layers
- Pooling layers
- Activation functions
- Linearizations
- No of Epochs
- Loss function
- Optimiser

Over the course of the project, multiple models with different combinations of the above parameters were used for training.

CNN Architectures



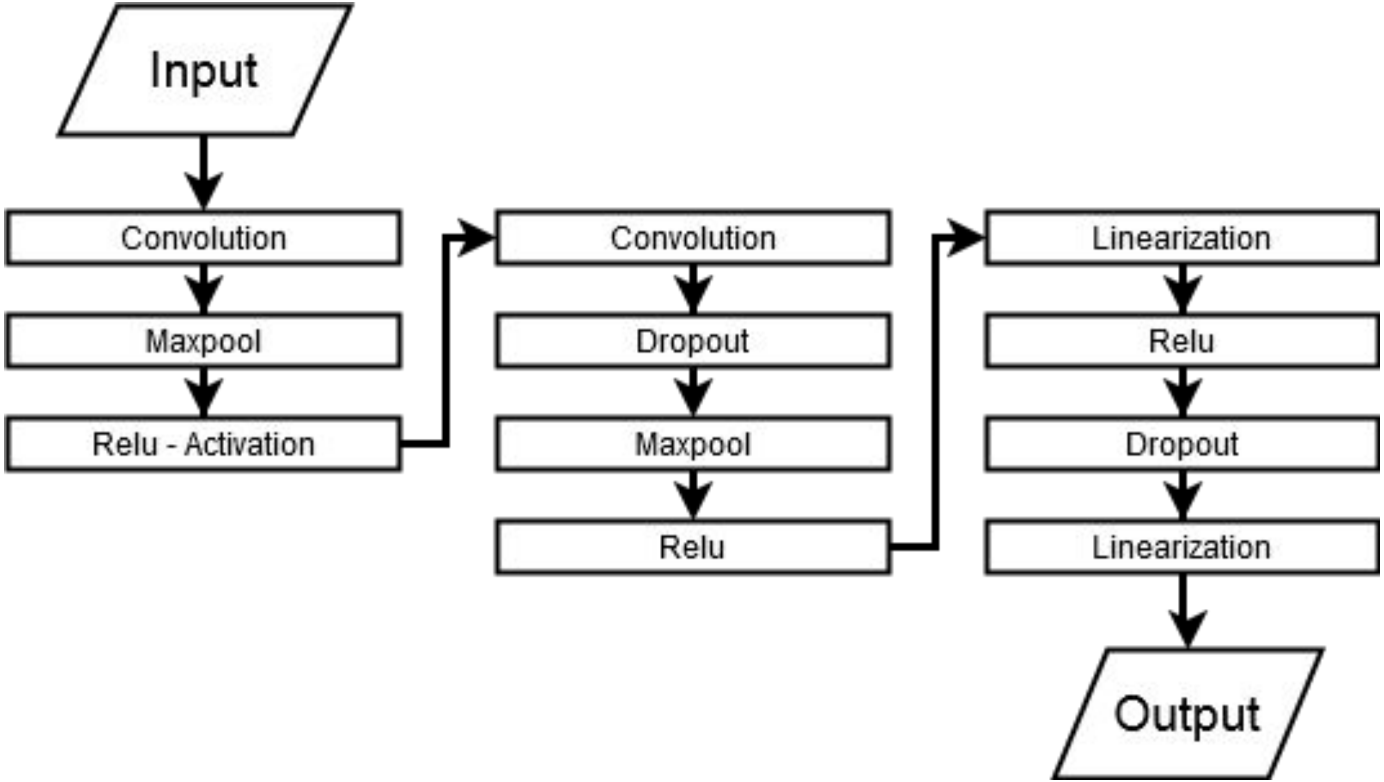
Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
VGG16	528 MB	0.713	0.901	138,357,544	23
InceptionV3	92 MB	0.779	0.937	23,851,784	159
ResNet50	98 MB	0.749	0.921	25,636,712	-
Xception	88 MB	0.790	0.945	22,910,480	126
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
ResNeXt50	96 MB	0.777	0.938	25,097,128	-

The top-1 and top-5 accuracy refers to the model's performance on the ImageNet validation dataset.

Depth refers to the topological depth of the network. This includes activation layers, batch normalization layers etc.

Source: https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d

Model 1 - Feature Extraction

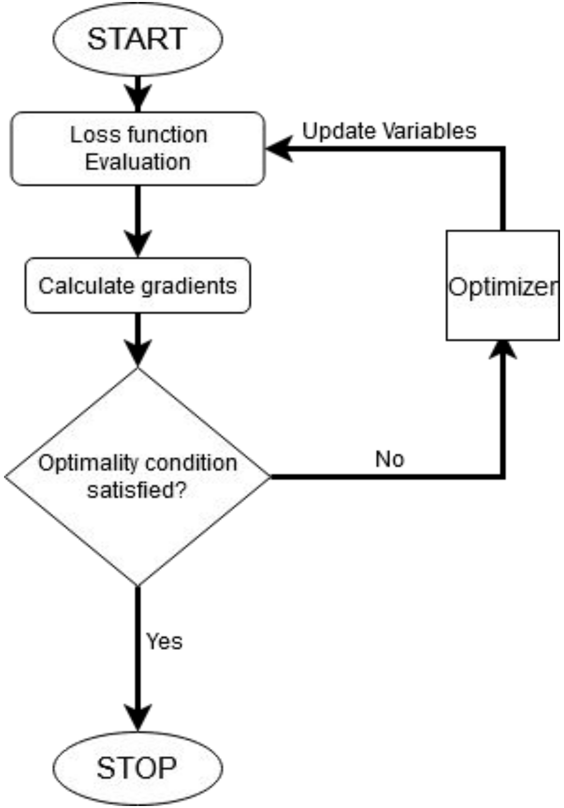


Model 1 - Feature Extraction

- In Convolution layer 1 : We give RGB (3), [32x32x3] inputs and we extract 10 features. As the filter kernel size is 3, the output would be [30x30x10].
- In Max Pooling 1 : With stride=2, [30x30x10] becomes [15x15x10].
- In Convolution layer 2 : With the filter kernel size is 3, [15x15x10] becomes [13x13x20].
- In Max Pooling 2 : With stride=2, [13x13x20] becomes [6x6x20].

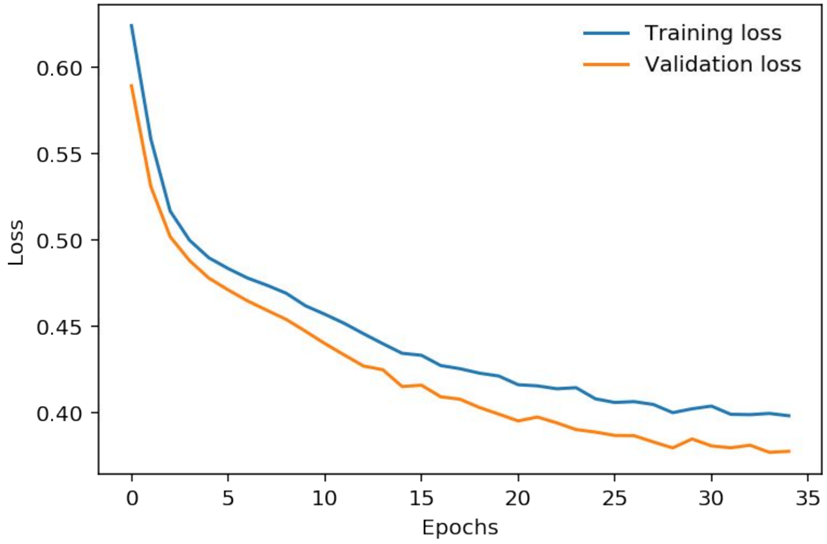
We shape this output into an array of 720 and apply linearization to an array of 1024 [32X32] which is then linearized to 2 outputs which correspond to the probability of it being in either of the 2 classes.

Model 1 - Loss and Optimisation



Loss function : Cross Entropy

Optimizer : Adaptive Moment Estimation (Adam)



Model 1 : Learning Curve

The validation loss is smaller than the training loss, but theoretically it should be lesser than training loss.

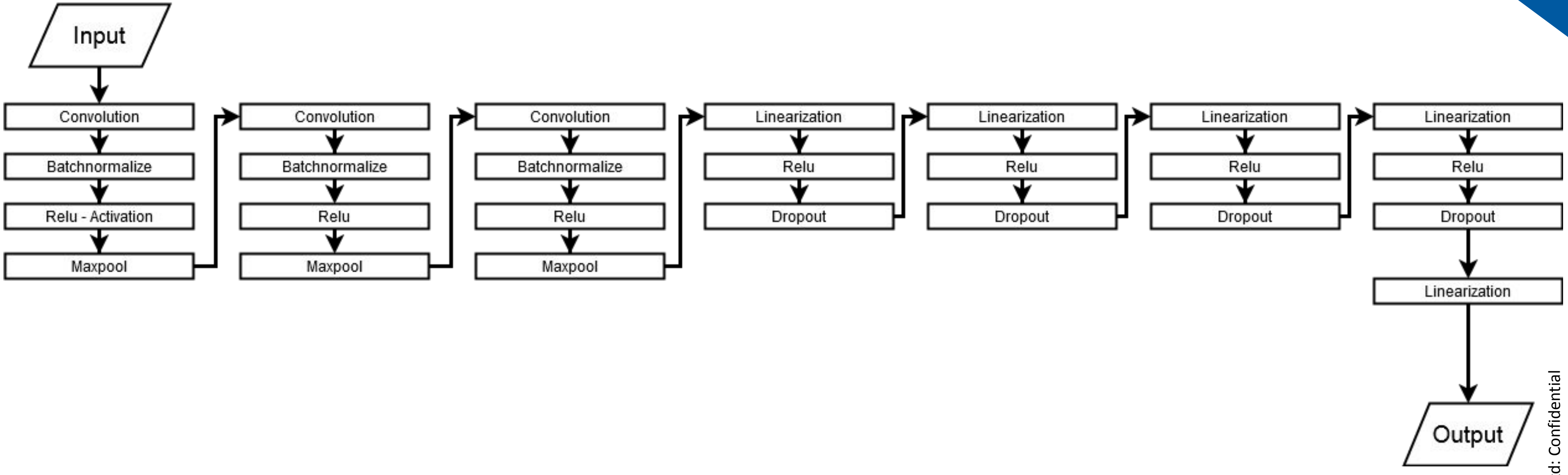
Reasons :

- Regularization is applied during training, but not during validation/testing. Training takes place in model.train() while testing takes place in model.eval() where regularisation functions such as dropout are inhibited.

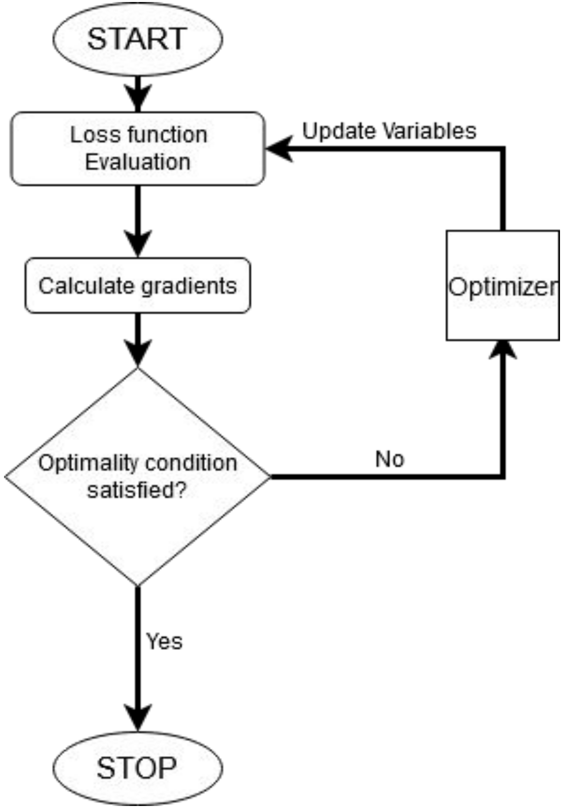
- Training loss is measured during each epoch while validation loss is measured after each epoch. On average, the training loss is measured 1/2 an epoch earlier.

- Your validation set may be easier than your training set. validation set may not have the same distribution (and difficulty) as your training set.

Model 2 - Feature Extraction

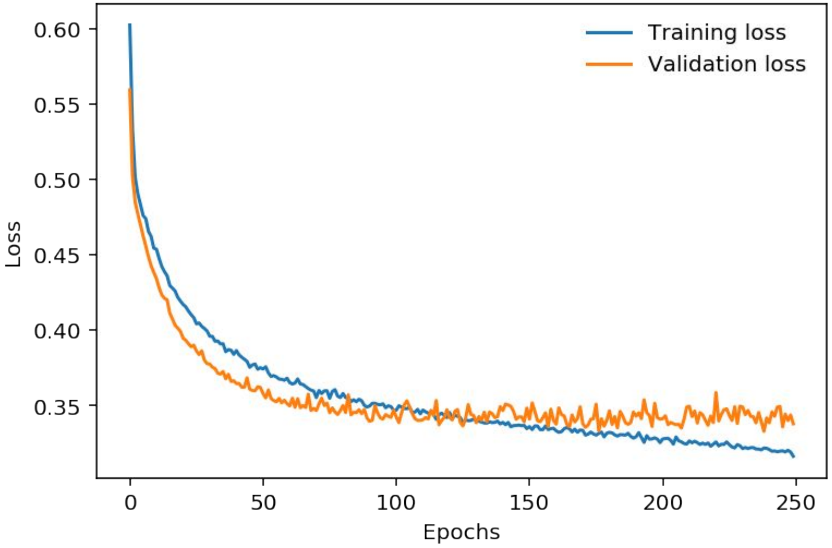


Modal 2 - Loss and Optimization



Loss function : Cross Entropy

Optimizer : Stochastic Gradient Descent (SGD)



Model 2 : Learning Curve

As the model is run for 250 epochs, the validation loss starts being significantly larger than training loss after 150 epochs.

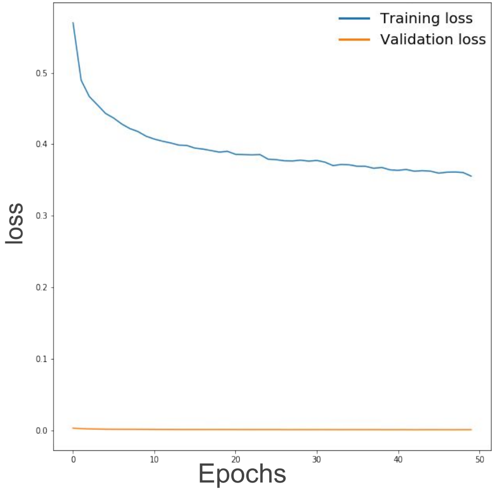
Reasons :

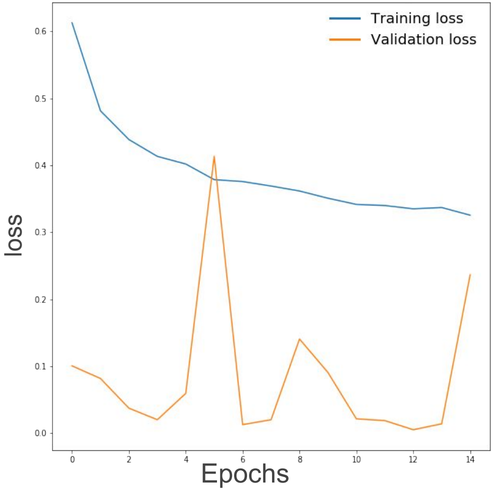
- The model is starting to over fit after 100 epochs. Here the model starts memorising the training set and keeps reducing the training loss, but any image not in the training set gives higher loss.

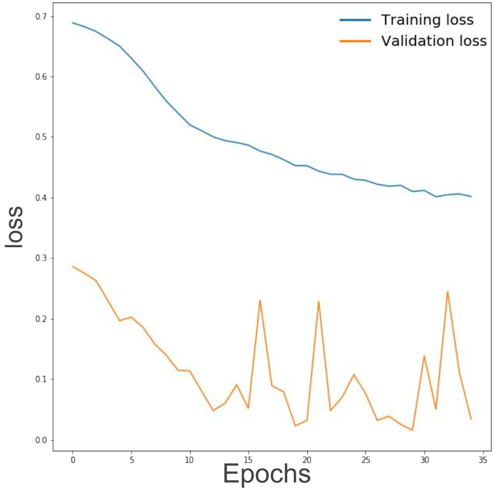
  This will further lead to the model being able to correctly classify only the images in the training set correctly.

Hence the model will be re-initialised and trained till 100 epochs for more accurate results.

Graphs I had got with earlier models







These graphs have Unrepresentative Validation Dataset. This issue was solved by increasing the size of the validation set.

Conclusion

	Model 1	Model 2
No of CNN layers	2	3
No of Features extracted	20	32
No of linear layers	1	5
No of dropouts	2	4
No of Epochs	35	100
Learning rate	0.001	0.001
Samples used for Testing	4710	4710
Correctly predicted in Testing set	3862	4003

Percentage White Analysis

Percentage white(Plane) to pass as a plane image, do differentiate between Partial Plane images and Plane images.

PLANE

PARTIAL PLANE

The percent White : 70.02% to pass as a plane image

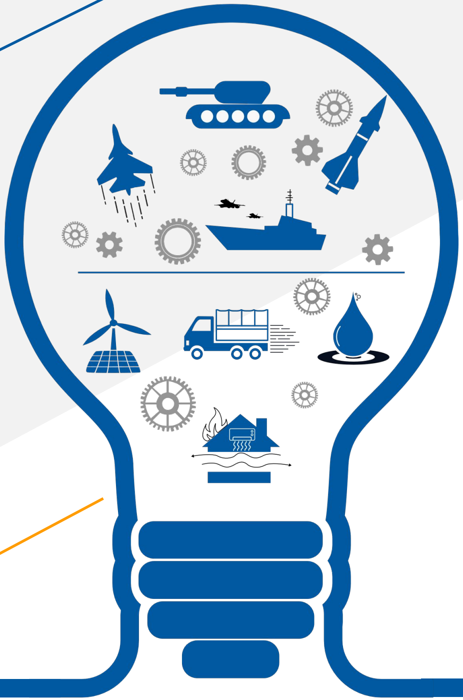
Percentage White Analysis

When the images are converted to Greyscale, each pixel would be between 0 (black) to 255 (white). We have taken three cases where we consider a pixel to be white if it is 60 percent, 70 percent and 90 percent of the grey scale.

In each case we found the percentage of the image that needs to be white for the image to be classified as a plane.

Percentage white in image to pass as plane images :

- With white percentage in grey scale as 0.6 scale gives 91.947% white required.
- With white percentage in grey scale as 0.7 scale gives 70.97 % white required.
- With white percentage in grey scale as 0.9 scale gives 30.116% white required.



Target Classification Of Ships

Classes : Cargo, Tankers, Military, Cruise, Carrier

Ship Images Database

Multi-Class Classification :

CARGO

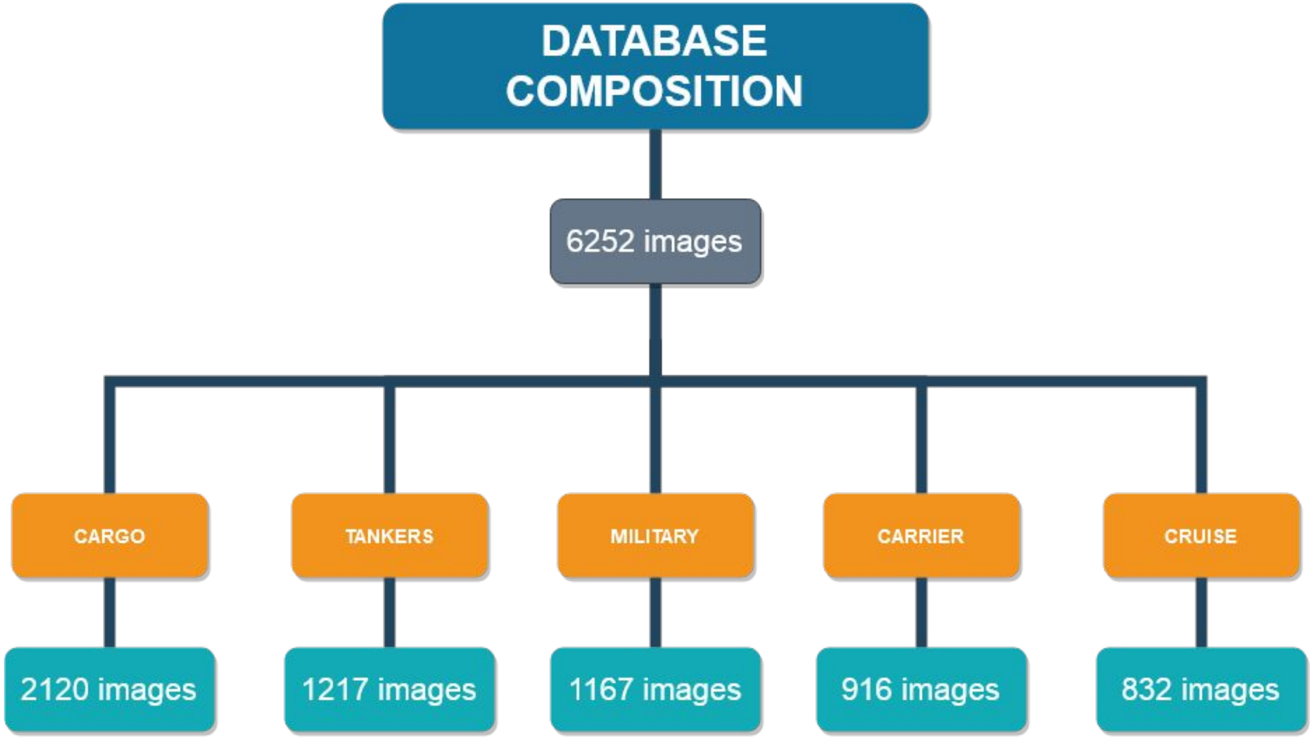
MILITARY

CRUISE

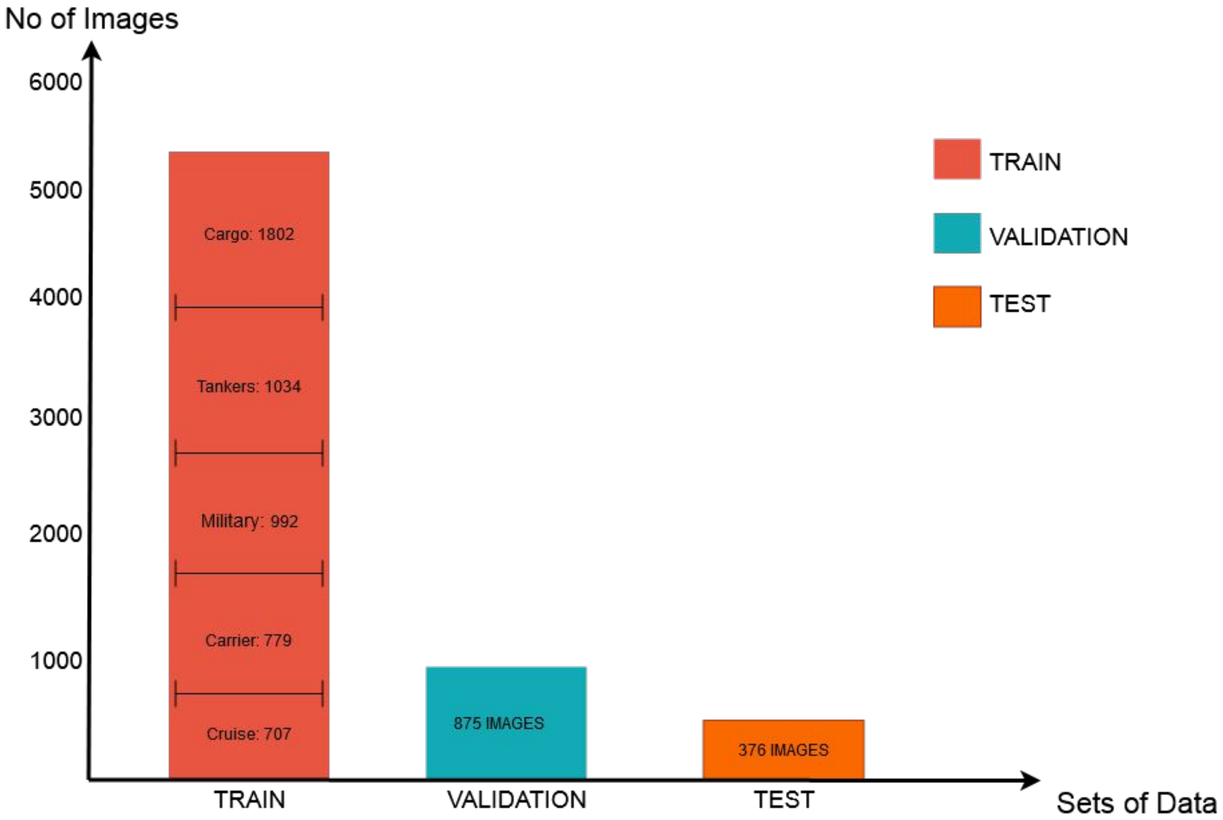
TANKER

CARRIER

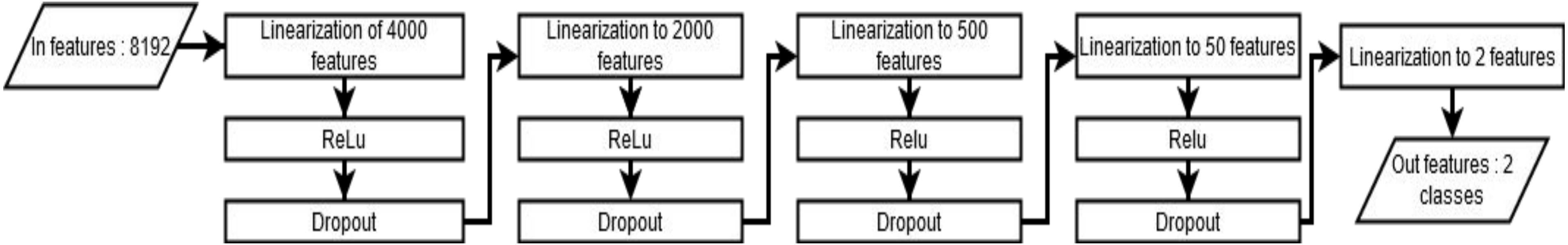
Ships database



Data Distribution



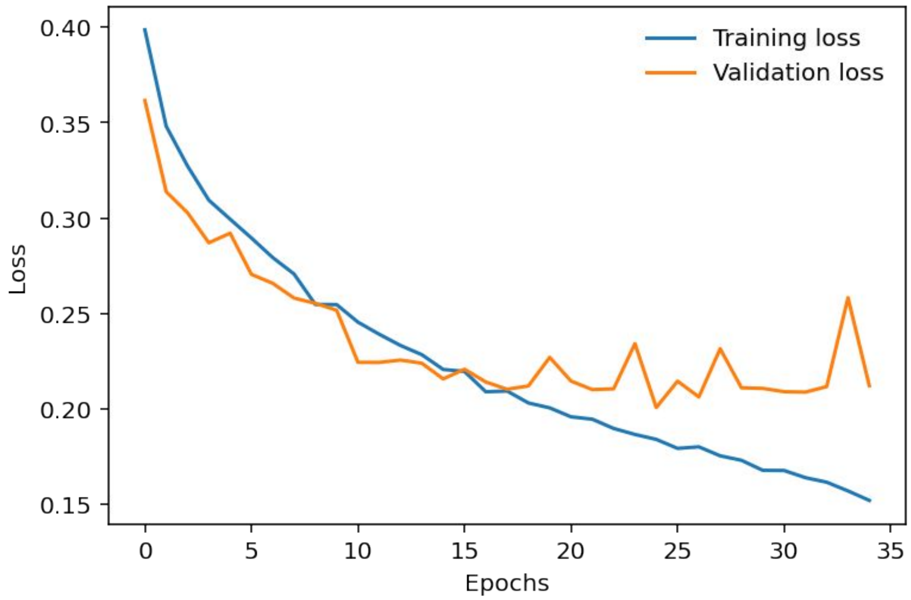
Comparison : Model 2 FC variants

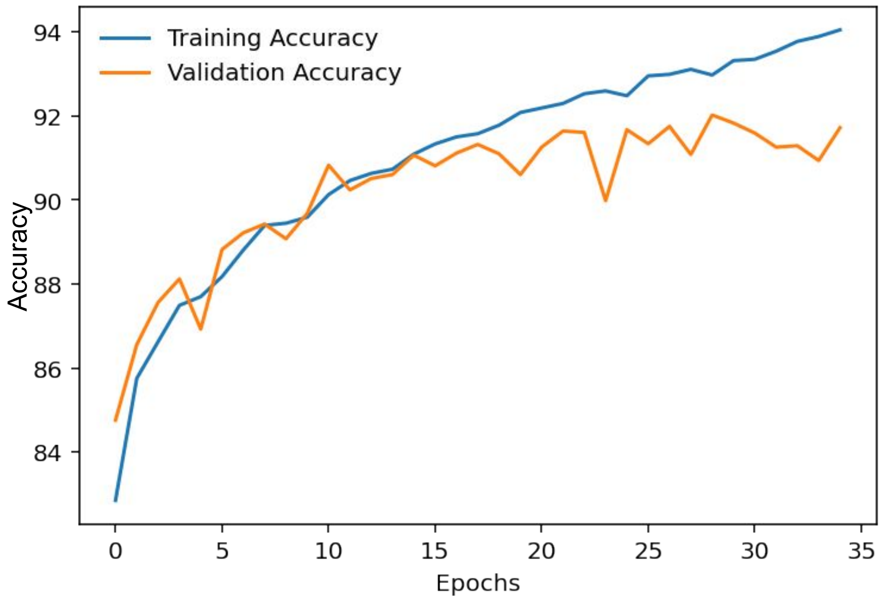


This is the model as is it. We are going to further reduce and increase the number of dropouts(regularisation) to see how our learning changes correspondingly.

Comparison : Model 2 FC variants

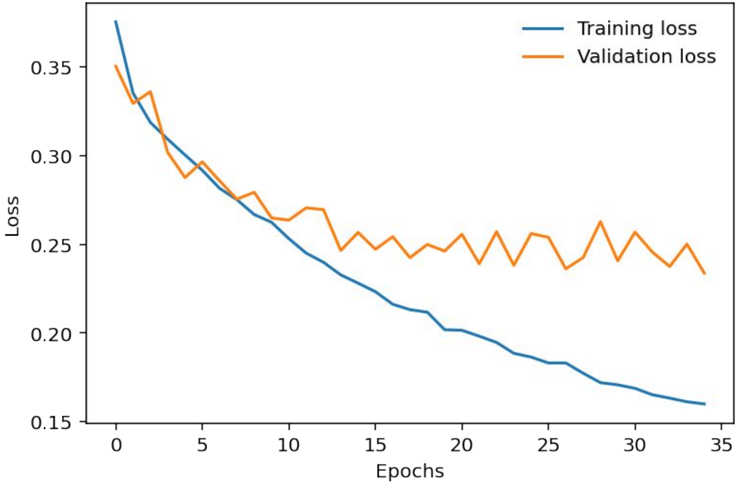
Model 2 learning curves :

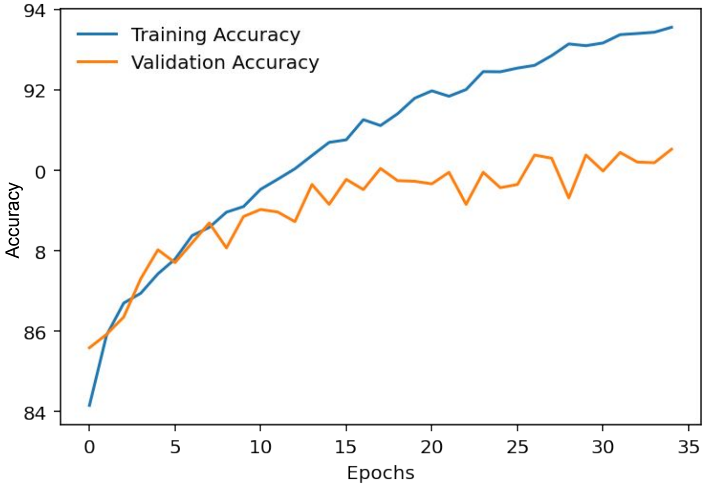


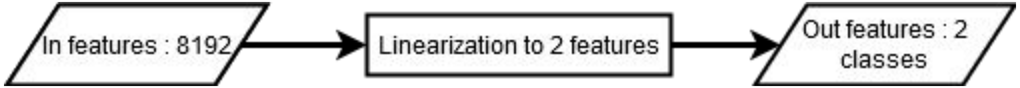


Comparison : Model 2 FC variants

Model 2 with no dropouts and activations in FC :

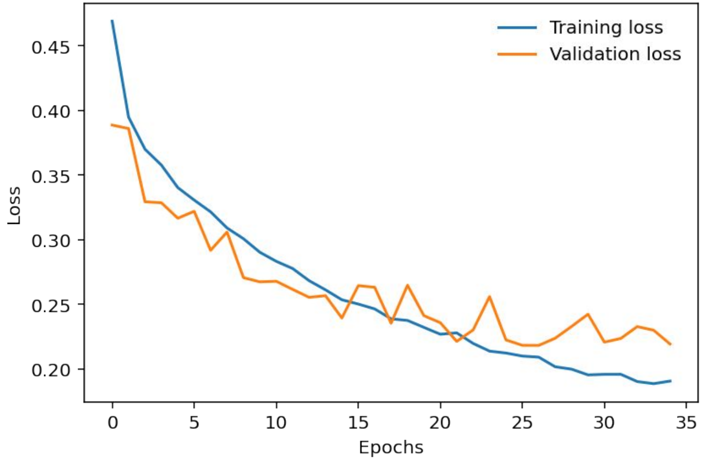


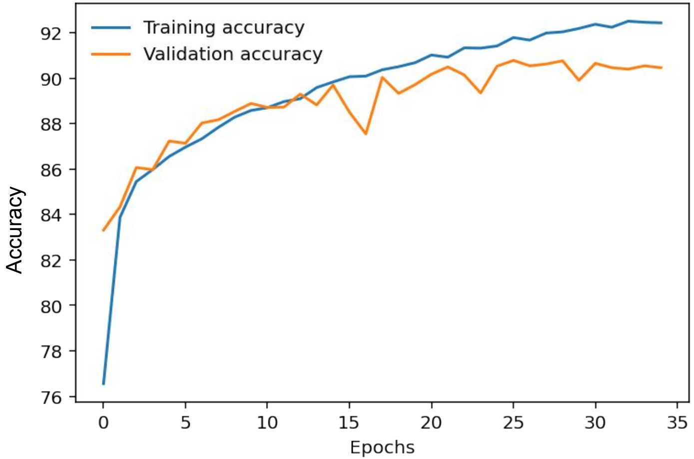


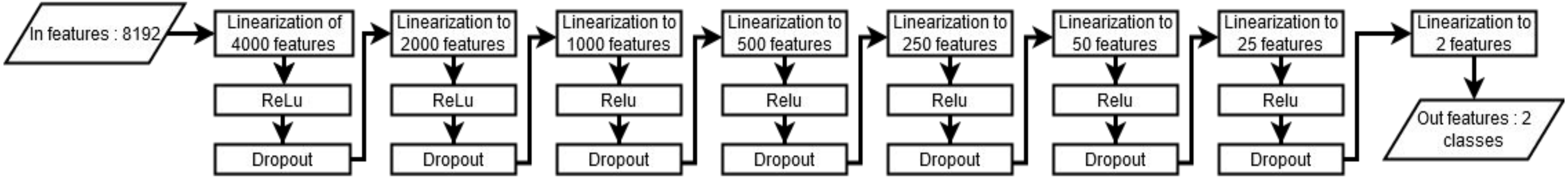


Comparison : Model 2 FC variants

Model 2 with more dropouts and activations in FC :





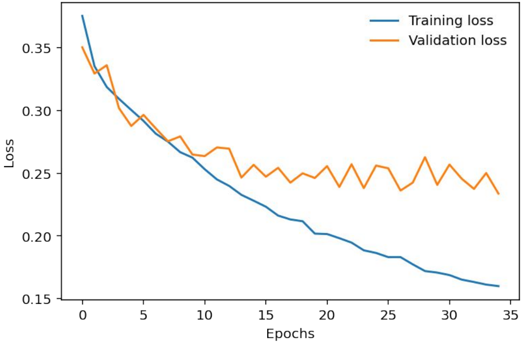


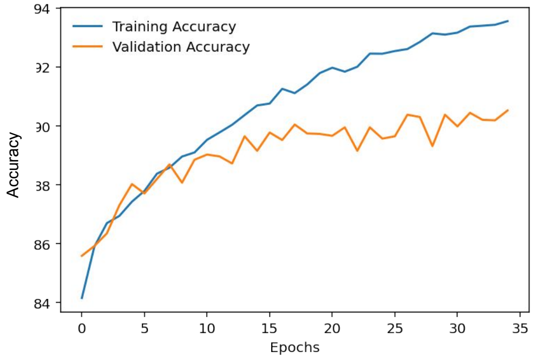
Comparison

	Model 2	Model 2 : No dropouts	Model 2 : More dropouts
No of linear layers	5	1	8
No of dropouts	5	0	8
No of Activations	5	0	8
Training Loss at the end	0.1554	0.1500	0.1943
Validation Loss at the end	0.2230	0.2400	0.2178
Training accuracy at the end	94	93.5	92
Validation accuracy at the end	92	90	91
Samples used for Testing	4710	4710	4710
Correctly predicted in Testing set	4003	3940	3973

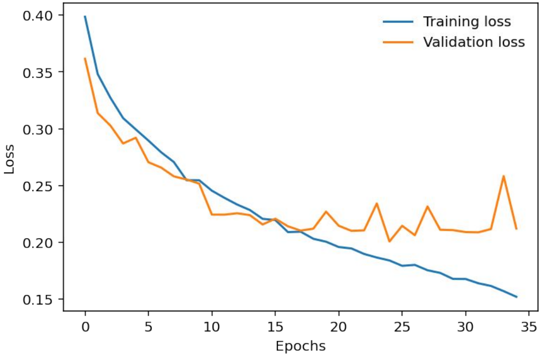
Comparison

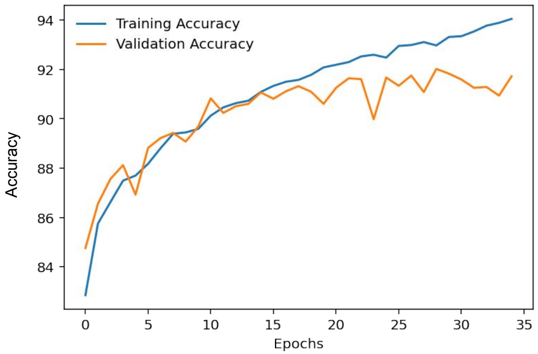
No Dropouts



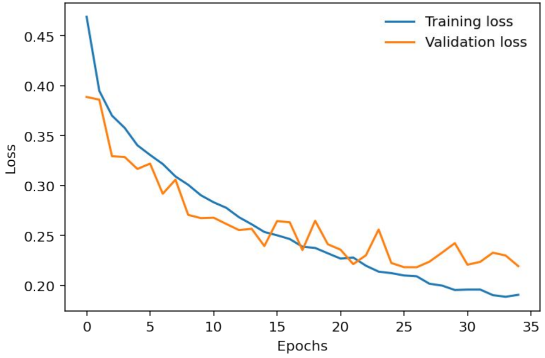


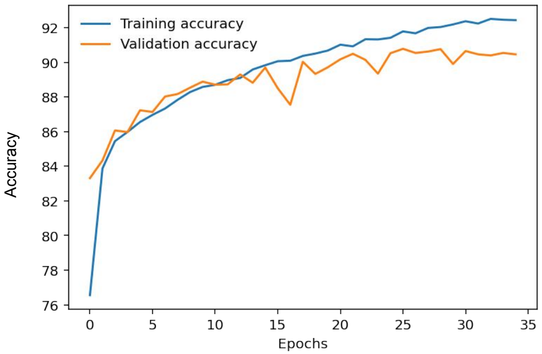
Normal Dropouts





More Dropouts





Transfer Learning

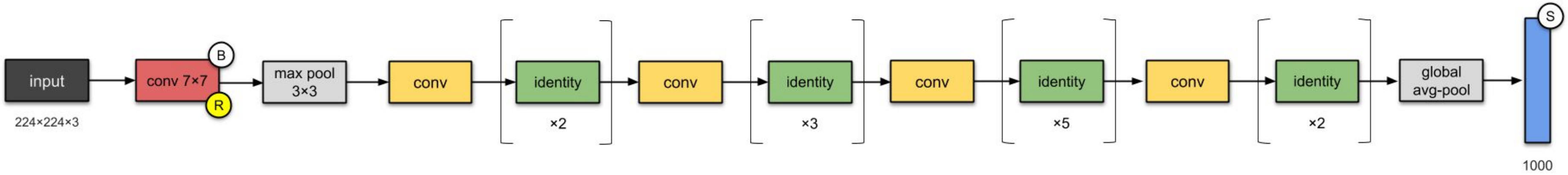
- This is the process in ML where we store knowledge gained while solving one problem and applying it to a different but related problem.

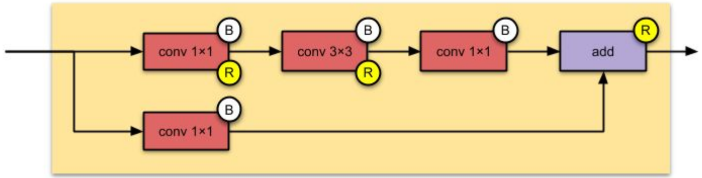
- In the case of Image Classification it is a technique where a neural network model that is already trained on other similar datasets and hence, we only need to carry out the remaining training in the last set of layers.

- For the Multiclass ship classification, RESNET50 is used. This is pretrained on the ImageNet database (it's a database containing millions of images belonging to more than 20,000 classes) .

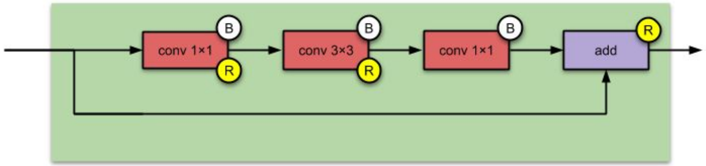
- The pretrained model has 2048 outputs, which we pass to our Fully-Connected layer to get 5 classe output.

ResNet50 Model





Conv block



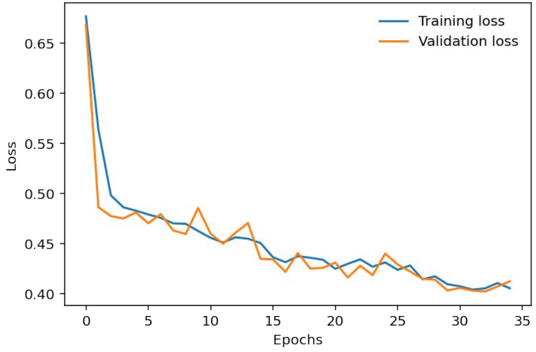
Identity block

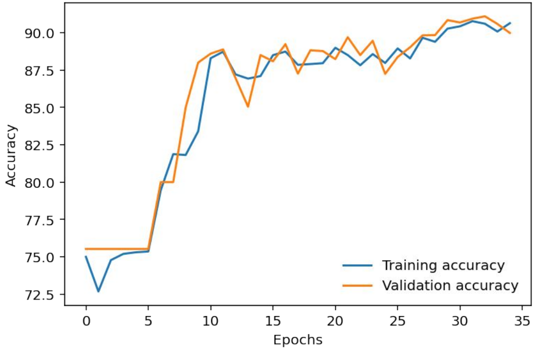
- The resnet model extracts 2048 features maps and we need to use fully connected layer to group them into 2 output classes of our image classification problem.
- The fully connected layer is computationally hectic as every node is connected to the other, hence to reduce the load and to increase generalisation, dropouts can be introduced.

Source: https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d

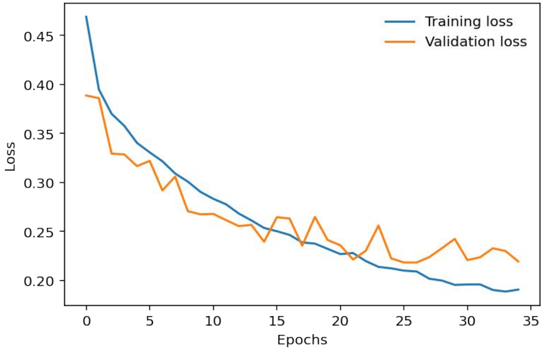
Comparison : Model 2 v/s Resnet50

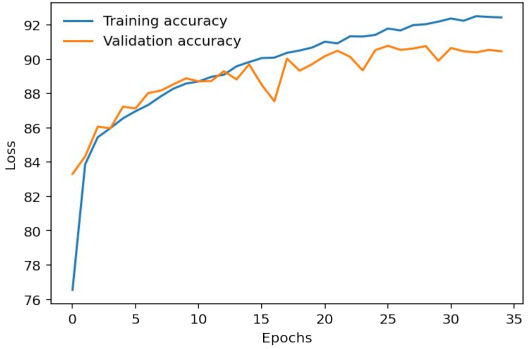
Resnet50:





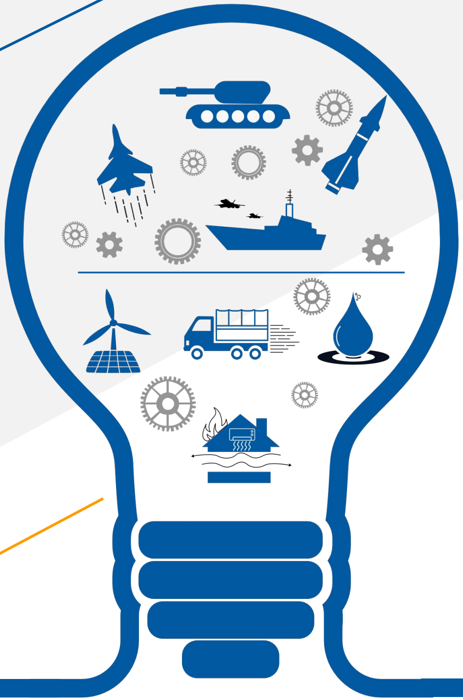
Model 2:





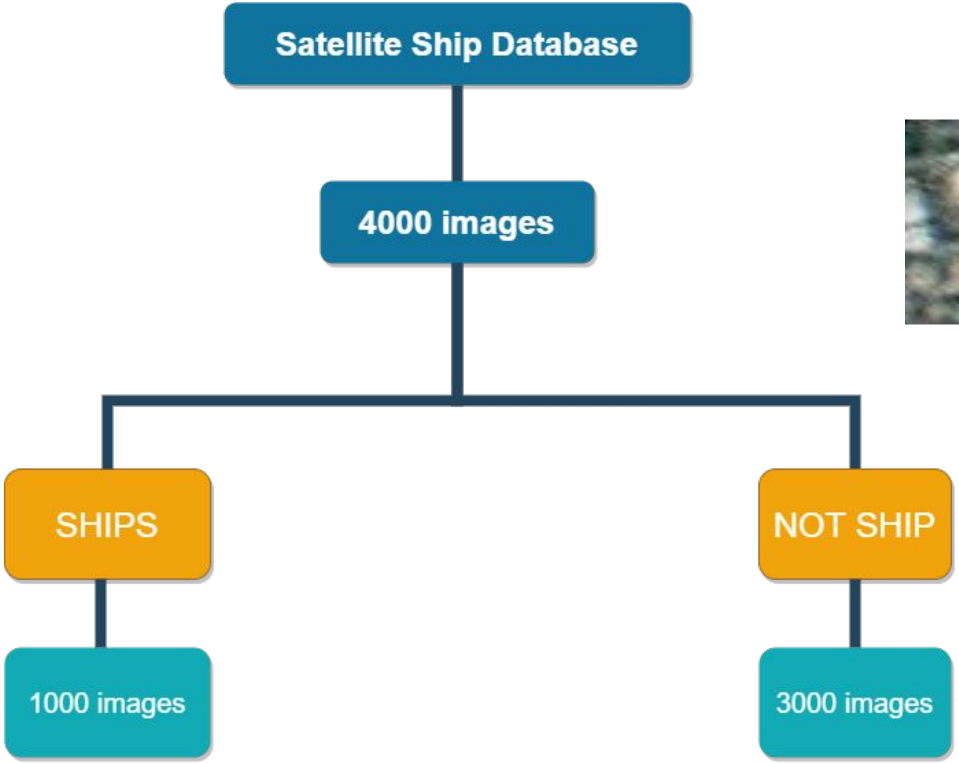
Comparison

	Model 2	Resnet50
No of CNN layers	3	50
No of Features extracted	32	2048
No of linear layers	5	8
No of dropouts	4	7
No of Epochs	35	35
Learning rate	0.001	0.001
Samples used for Testing	4710	4710
Correctly predicted in Testing set	4003	4145



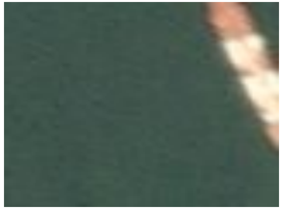
Classification of satellite images of Ships

Database : Labels and Distribution




Scenery

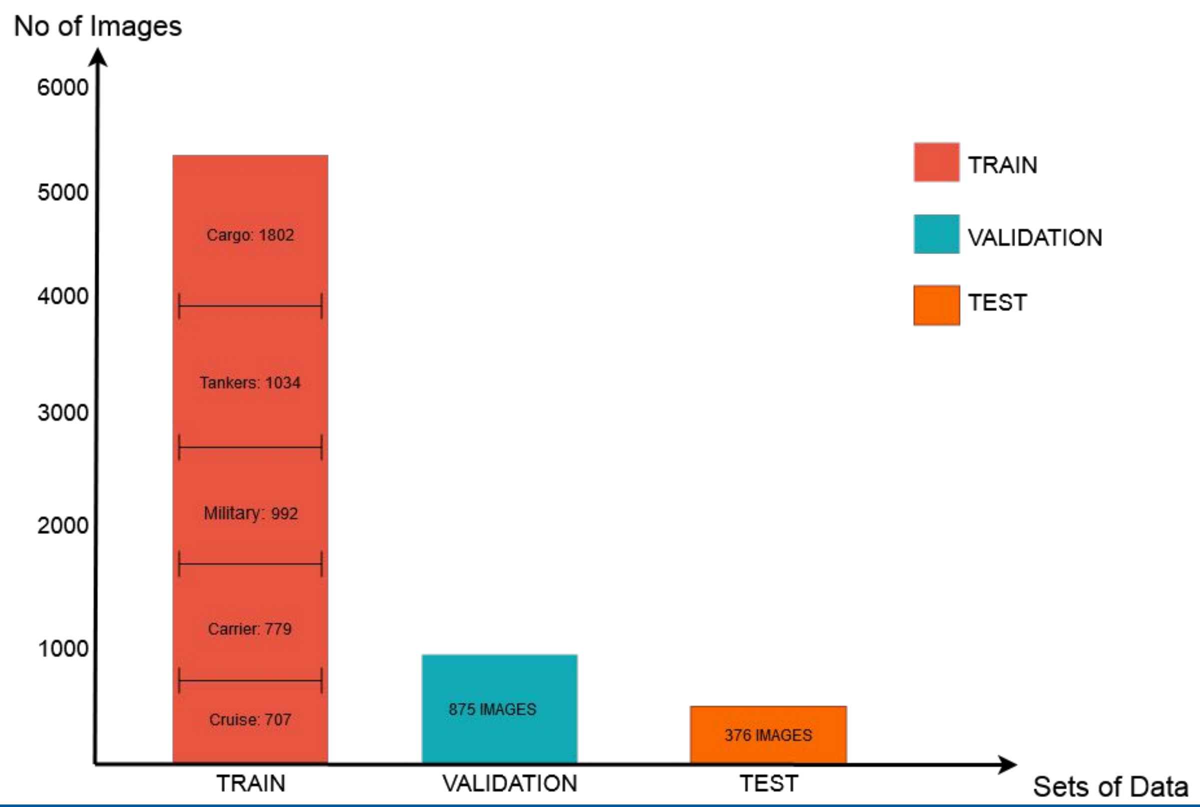

Confusers


Partial Ships

NON SHIP


SHIP

Data Distribution

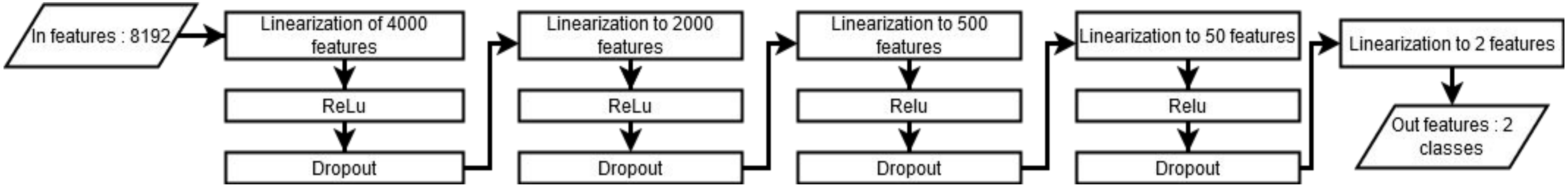


Hyperparameters

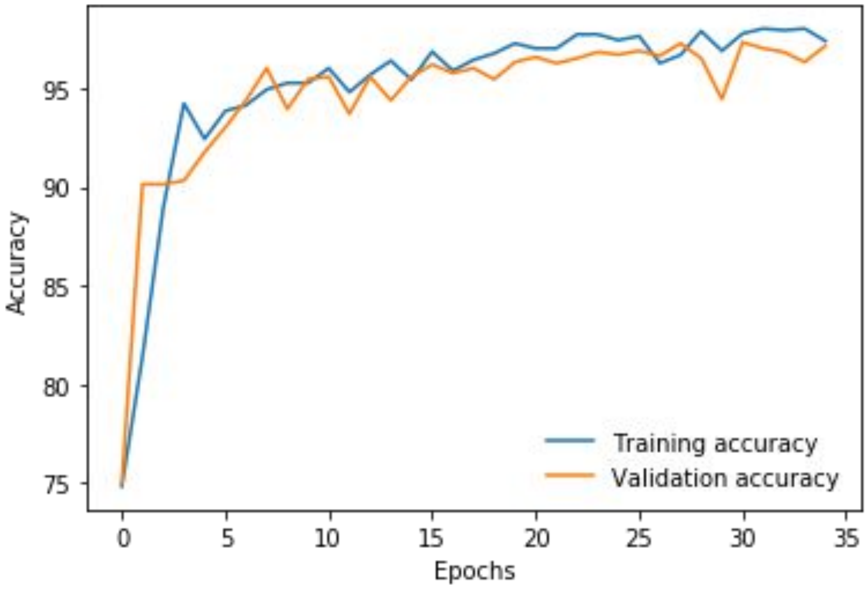
- No of epochs : 35
- Optimizer : Adam
- Batchsize : 25
- Train-test split ratio : 60:40
- Learning rate : 0.01

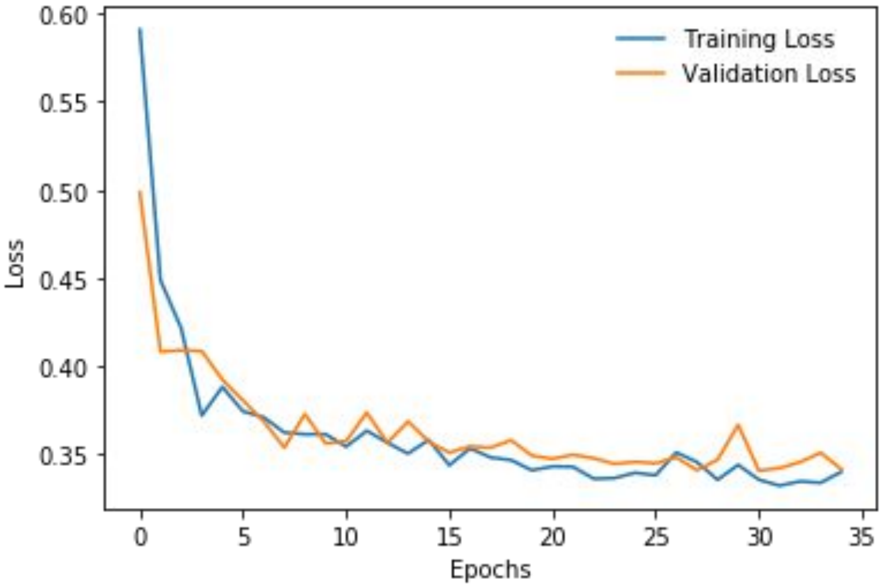
This classification exercise is very similar to the Binary Plane classification.
Model 2 with moderate regularisation is used.

Initial Model used(model 2)



Binary Classification

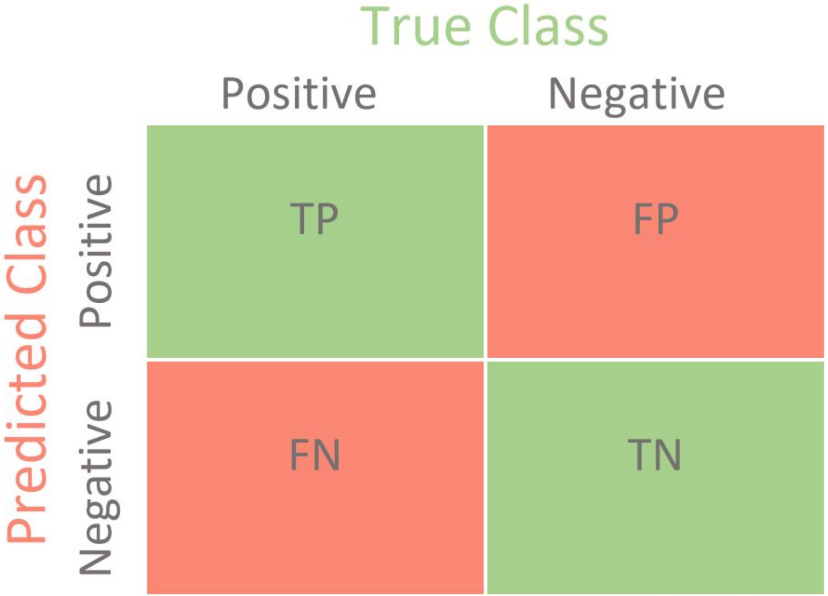




Binary Satellite Ship Classification : 97.12%

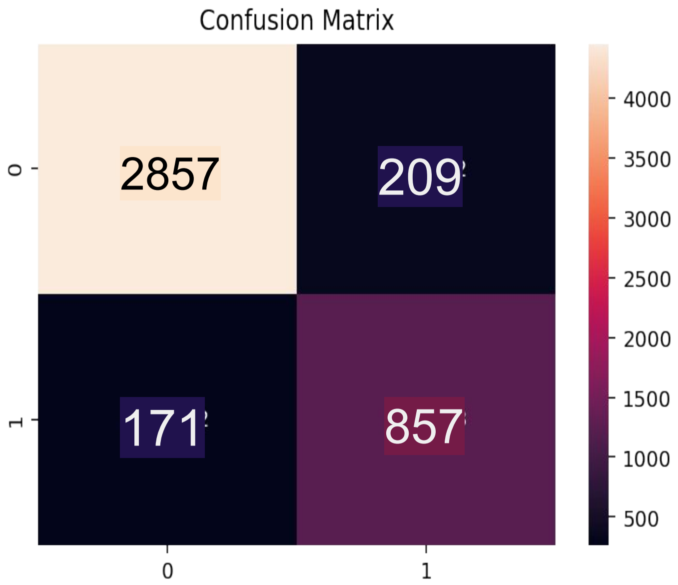
Comparison : Model 2 v/s Resnet50

Confusion Matrix : In the case of a binary classification, it gives the number of True Positives and True Negatives and False Positives and False Negatives.

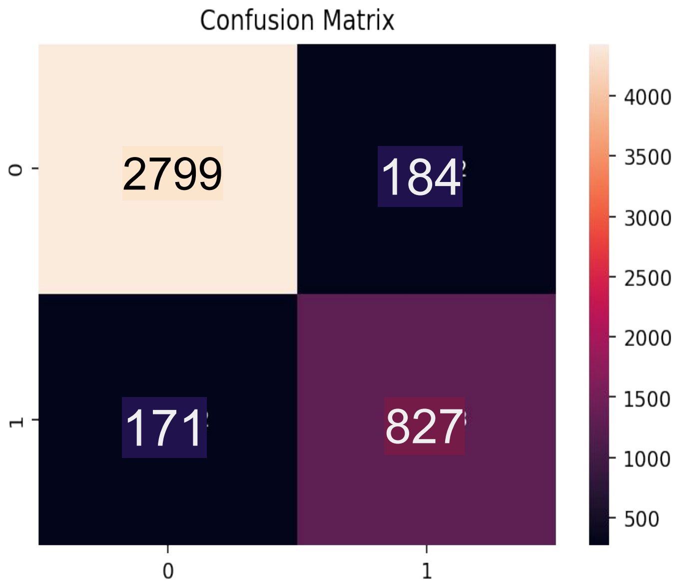


Source: https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model

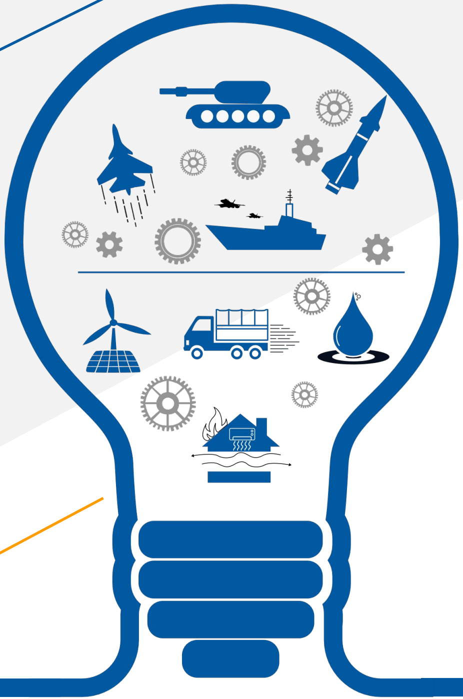
Comparison : Model 2 v/s Resnet50



Model 2



Resnet50



High Vehicle classification

Binary and Multi-class

Images and Classes

VEHICLE

Car

Bus

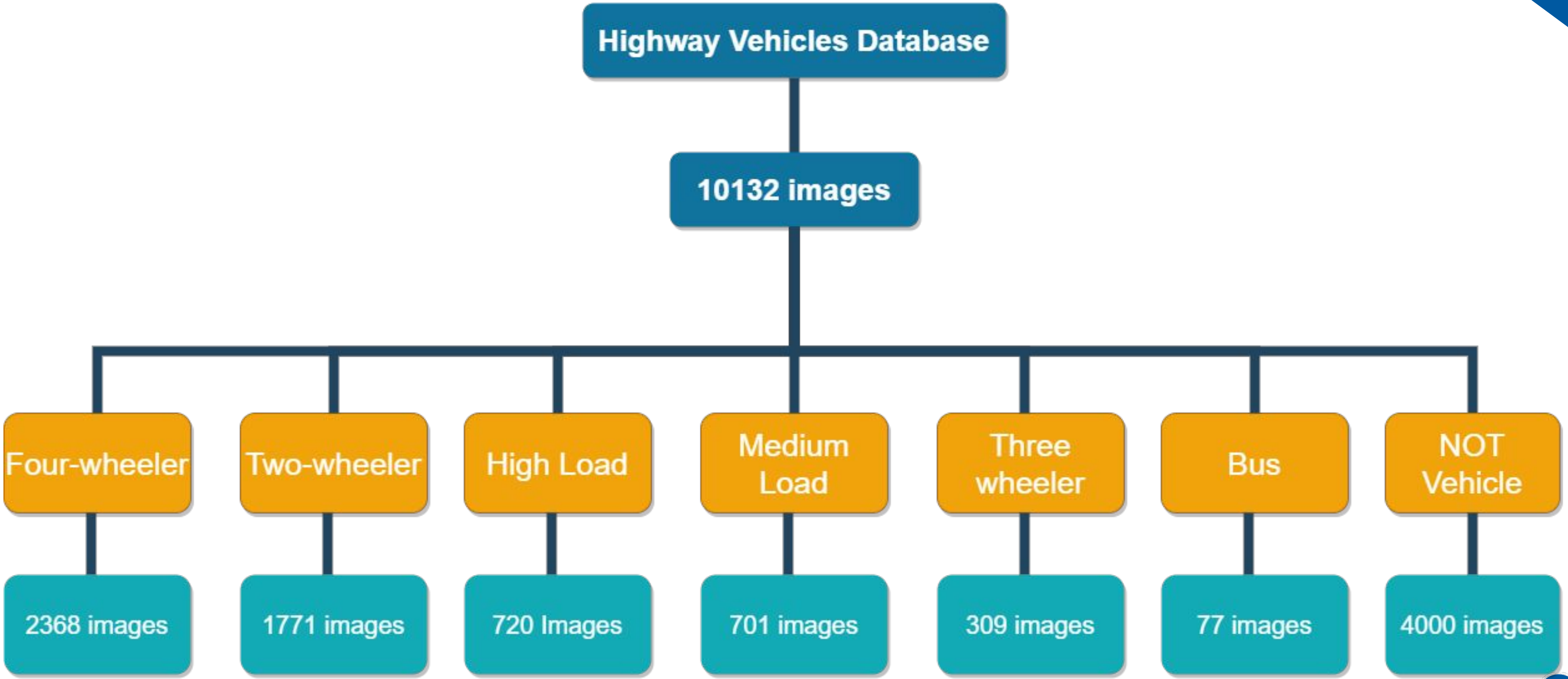
Heavy Load

2-Wheeler

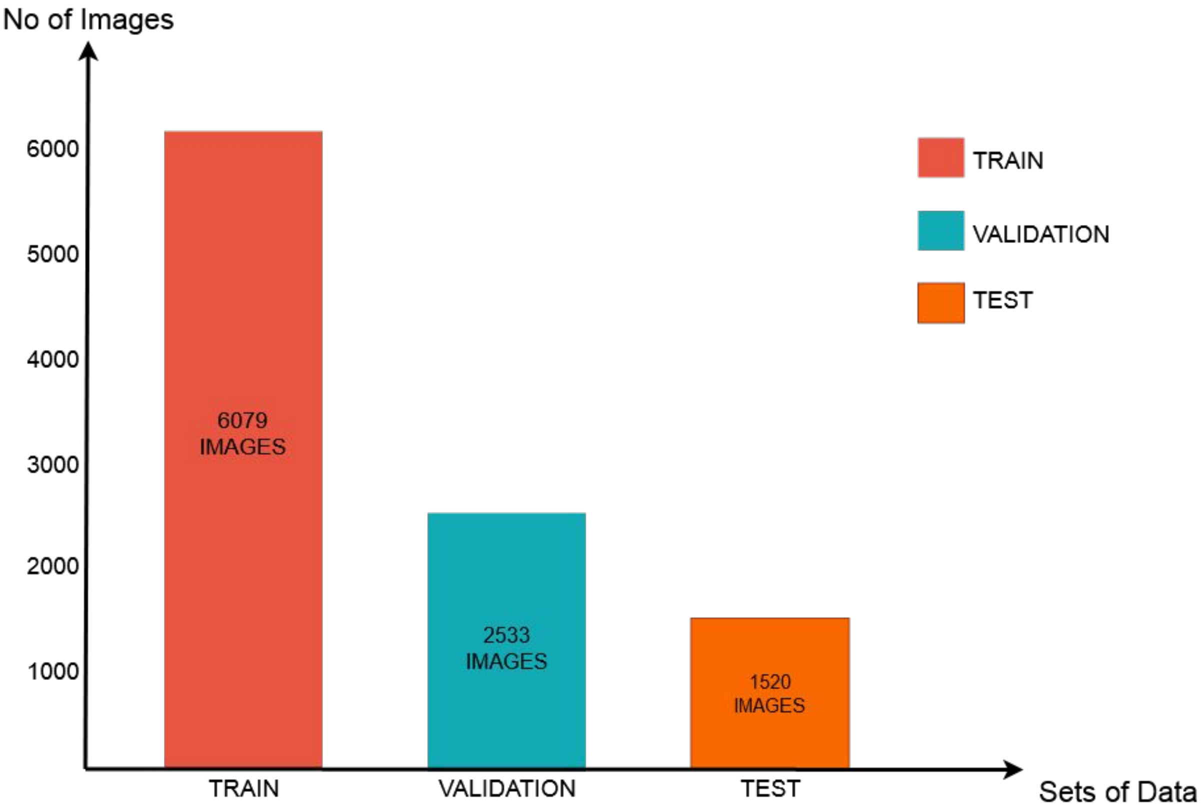
Medium Load

NON-VEHICLE

Database : Labels and Distribution



Data Distribution in Sets



Working with the data.

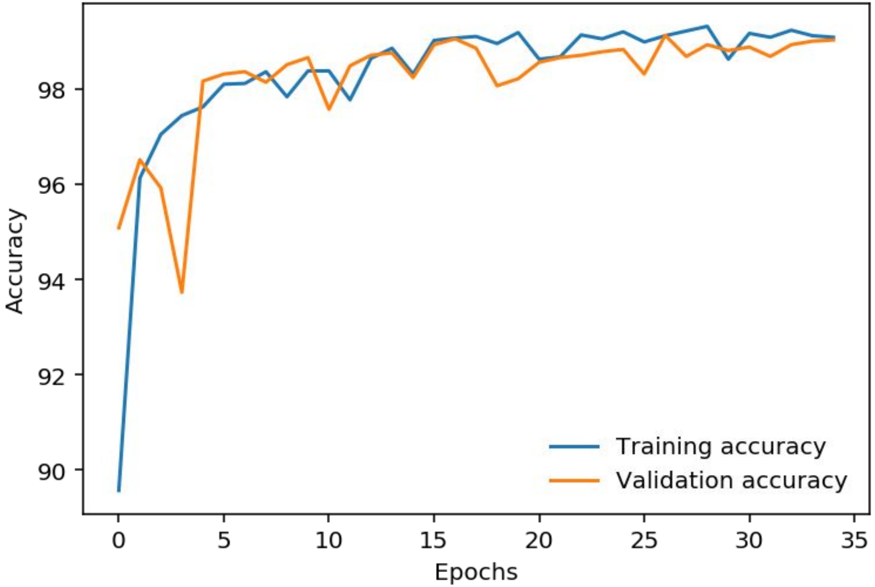
- The main issue when working with Highway Vehicle Data is that it has a lot of disturbance at the background. By this it means that the background has features that can be captured as image features very easily.
- One way to fix this is to use more dropout such that these features may be given less importance.
- The other way to fix this is to crop the image to have only the object of interest. If bounding boxes are not accessible, then we can use a Canny Edge detector to find major deflection of edges to find the edge of our desired object.

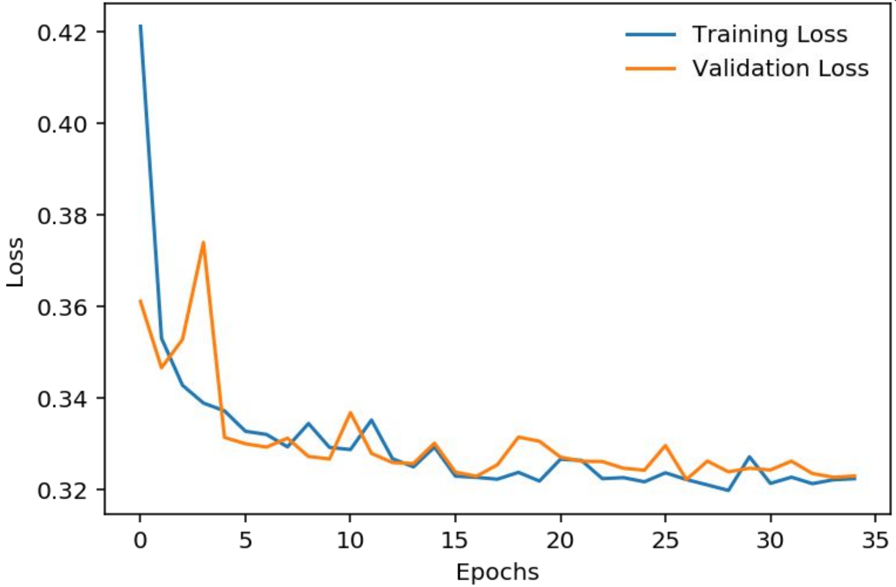
Hyperparameters

- No of epochs : 35
- Optimizer : Adam
- Batchsize : 100
- Train-test split ratio : 60:40
- Learning rate : 0.01 with cosine annealing

This classification was carried out with lesser number of dropouts as the learning was difficult due to the similarities in the different types of vehicles on the highway.
The training was more difficult in the multiclass classification as the number of images in the some of the subclasses were possibly insufficient.

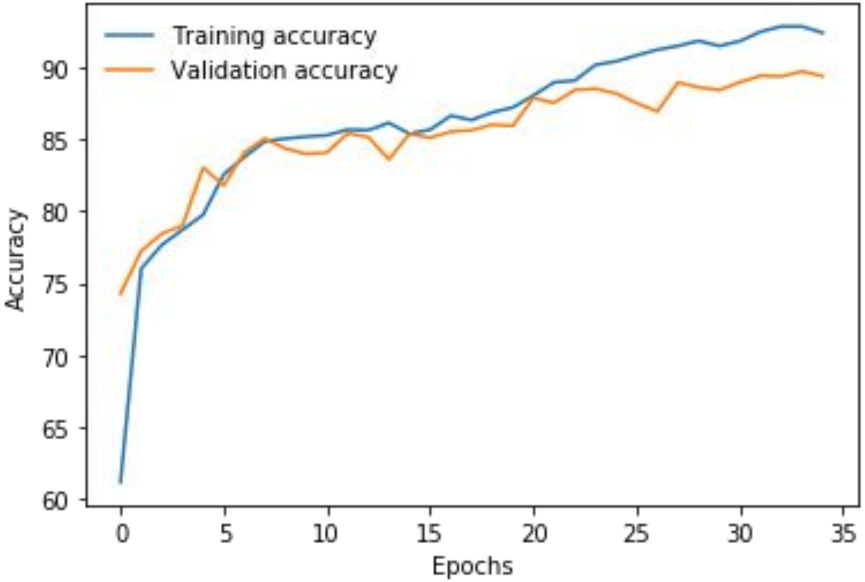
Binary Classification

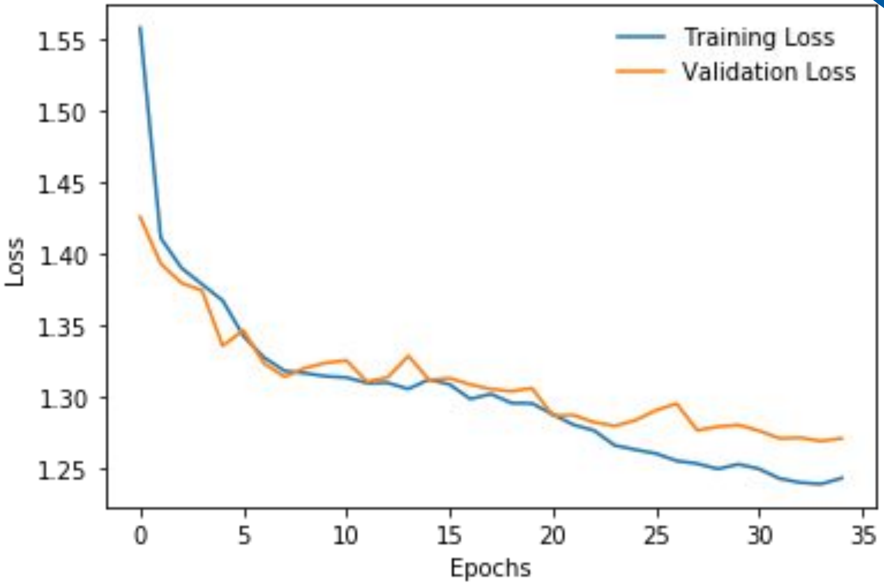




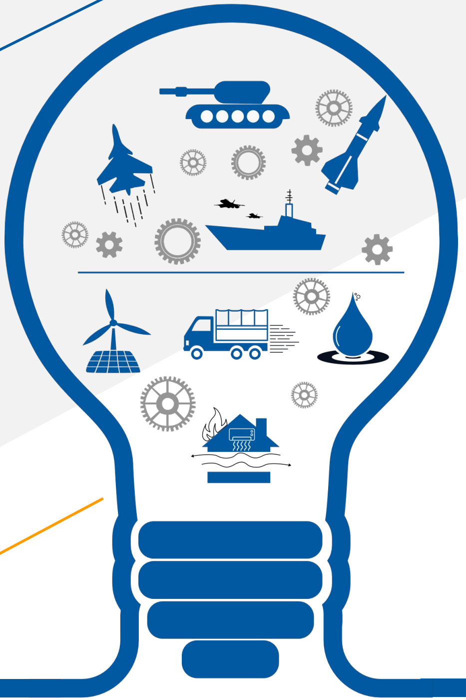
Binary Highway Vehicle Classification : 99.03%

Multi-class classification





Multiclass Highway Vehicle Classification : 89.41%



SARS image classification

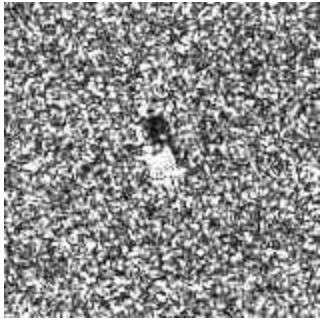
Multi-class : 2S1, BRDM_2, SLICY, ZSU_23_4 Tanks.

Synthetic-aperture Radar : SAR

- SAR is a form of radar that is used to create two-dimensional images or three-dimensional reconstructions of objects.
- SAR is typically mounted on a moving platform, such as an aircraft or spacecraft.
- Electromagnetic waves are transmitted sequentially, the echoes are collected and the system electronics digitizes and stores the data for subsequent processing.
- Rough surfaces appear brighter, as they reflect the radar in all directions, and more of the energy is scattered back to the antenna. A rough surface backscatters even more brightly when it is wet.

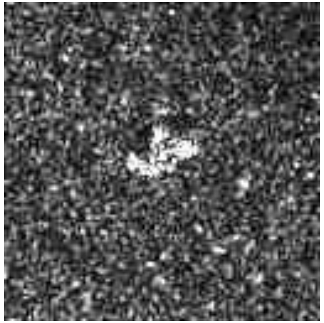
SARS Image Classification

Multi-class classification:

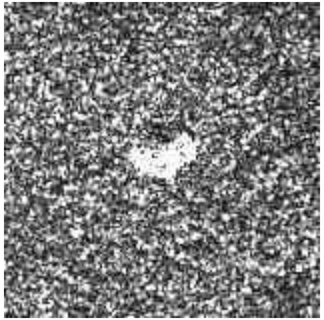


2S1

SLICY

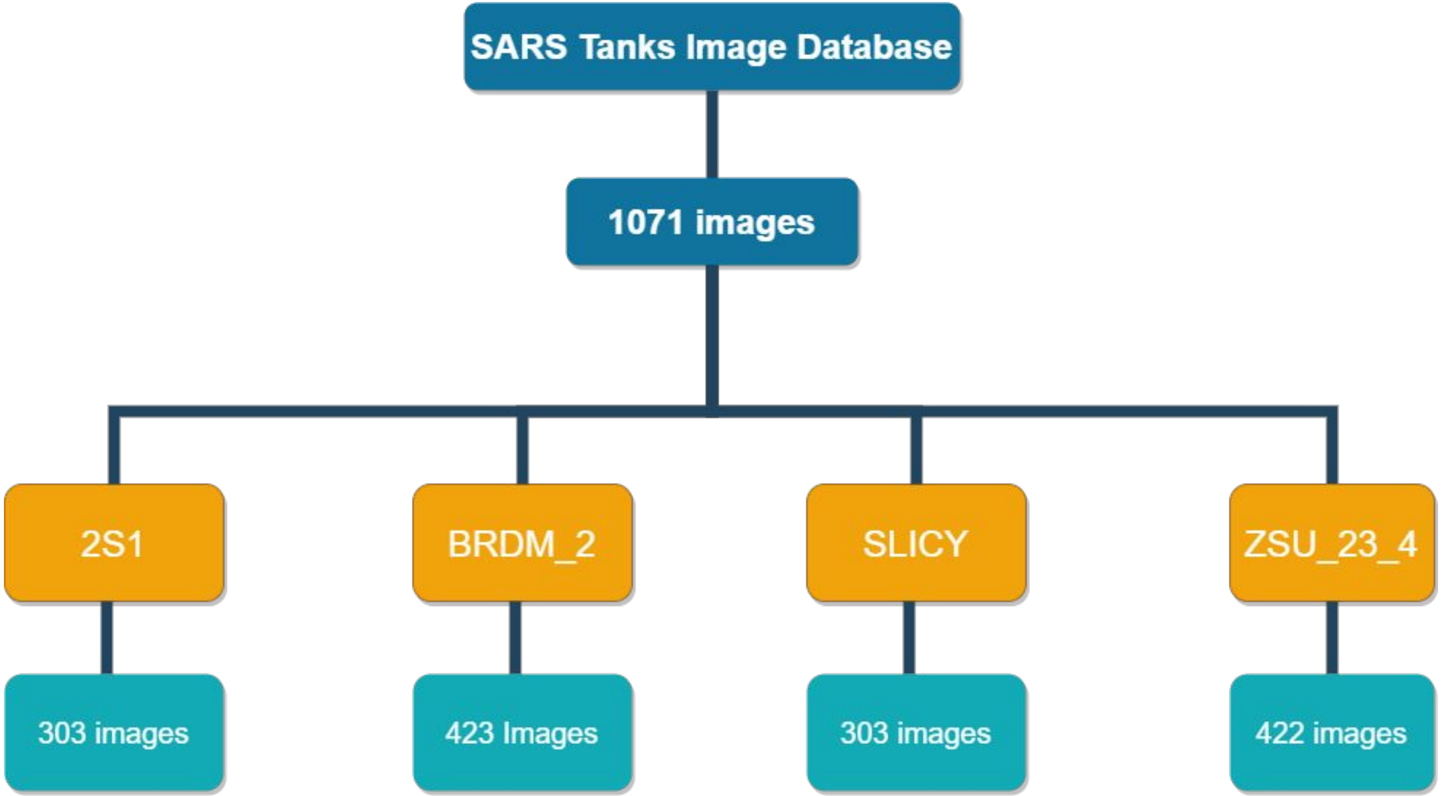


BRDM_2

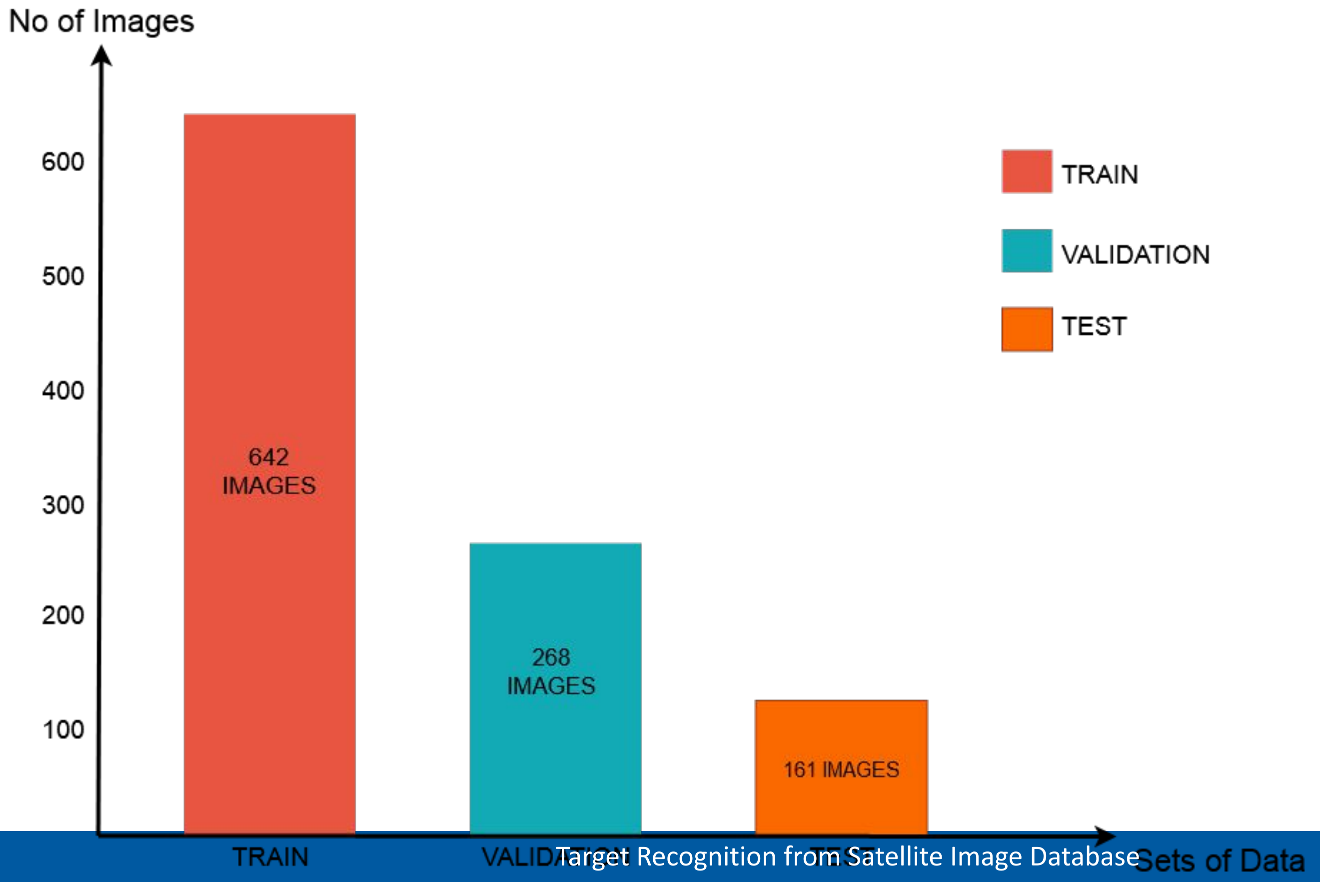


ZSU_23_4

Database : Labels and Distribution



Data Distribution in Sets

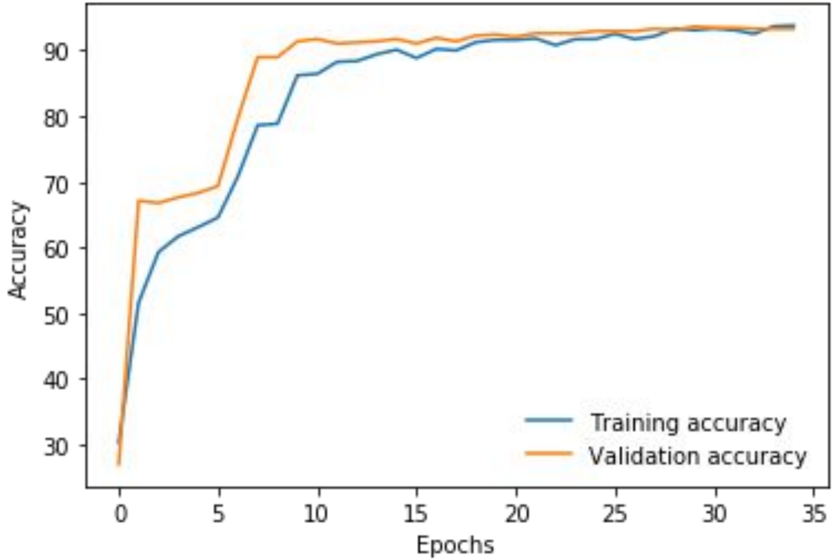


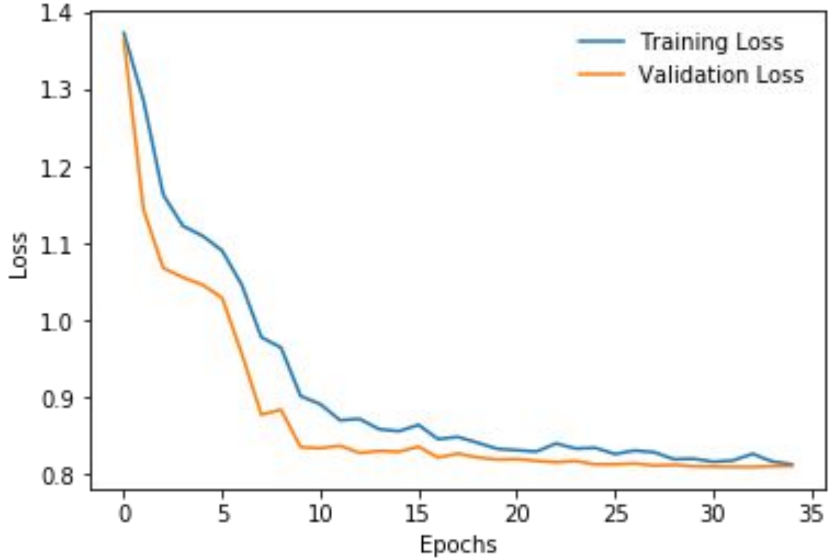
Hyperparameters

- No of epochs : 35
- Optimizer : Adam
- Batchsize : 25
- Train-test split ratio : 60:40
- Learning rate : 0.01 with cosine annealing

When the number of dropouts were altered, it was noticed that the learning saturates at a certain epoch and then resumes learning after a few epochs when more number of dropouts are used.
The possible reason for this is that some of the features which were necessary to be specifically classify were being dropped during the regularisations.

Multiclass Classification





Multiclass SARS Tanks Classification : 93.28%

Conclusion

- This process of data processing, defining various algorithms, cross validation, training, testing, selecting final algorithm and searching for reasons for low accuracy is a time-taking task.
- Every data is different, so if one is successful with one type of data, it doesn't necessarily mean that he/she will be successful on some other data

Different datasets classified for learning purpose with accuracies :

1. Binary Satellite Plane Classification : 91.24%
2. Binary Satellite Ship Classification : 97.12%
3. Multiclass Ships Classification : 90%
4. Multiclass Highway Vehicle Classification : 89.41%
5. Binary Highway Vehicle Classification : 99.03%
6. Multiclass SARS Tanks Classification : 93.28%

References

- https://towardsdatascience.com/feature-engineering-what-powers-machine-learning-93ab191bcc2d
- https://www.linkedin.com/pulse/machine-learning-model-performance-error-analysis-payam-mokhtarian
- https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/
- https://towardsdatascience.com/handwritten-digit-mnist-pytorch-977b5338e627

Thank You !

Rishav Kanth

+91 72760 31511

rishav.kanth@zeusnumerix.com

www.zeusnumerix.com