Deep Learning based Approach to Range Regression

V. Adithya Krishnan
Anurag Shukla
Samvram Sahu

Guided by - Dr. Deepak Mishra, Shri Pradipta Roy

AIM

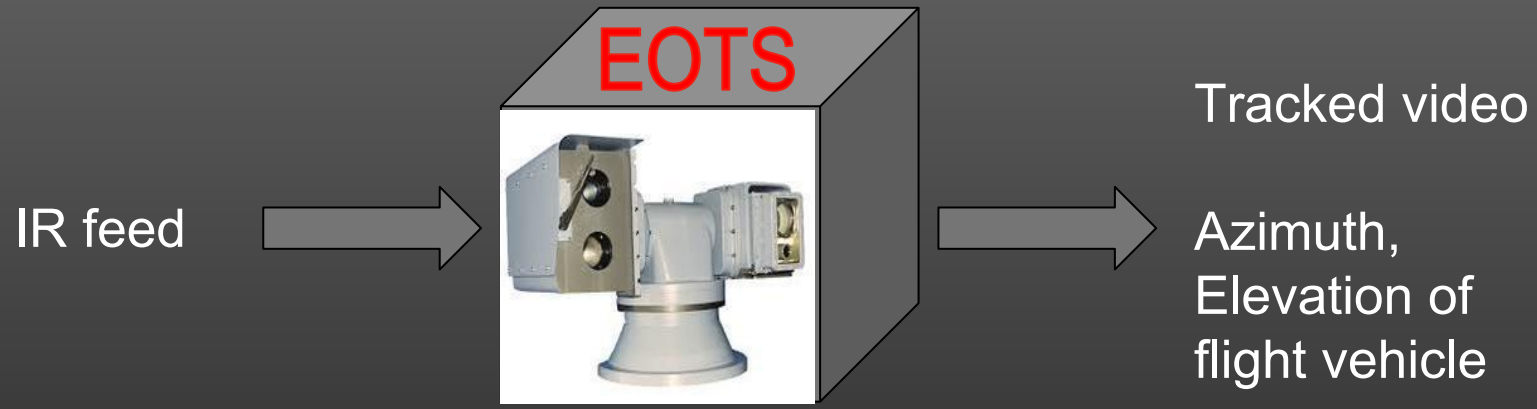
Given a **frame** depicting a missile in flight from an EOTS,

Predict the **range** of the flight vehicle within certain bounds of error.

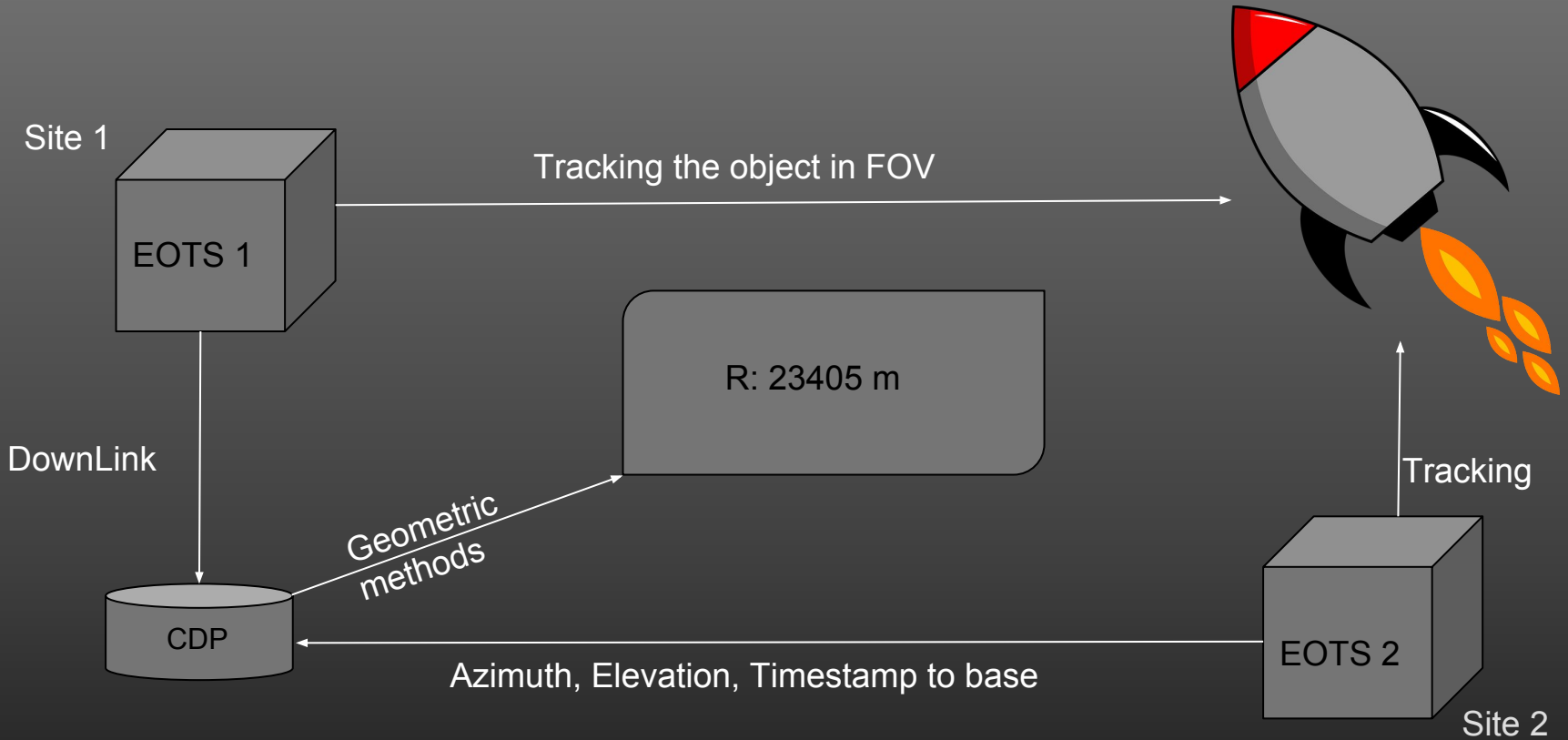
Relevance of the Topic

1. Absolutely new domain, we need to create our own dataset, hence novelty from the roots.
2. This is what the country's prime Defence organization prioritized.
3. From cleansing and pre-processing the data to establishing results, we provided a test-bed.
4. Blooming Area, DARE(Defence Avionics Research Establishment) is looking into our work as we speak.

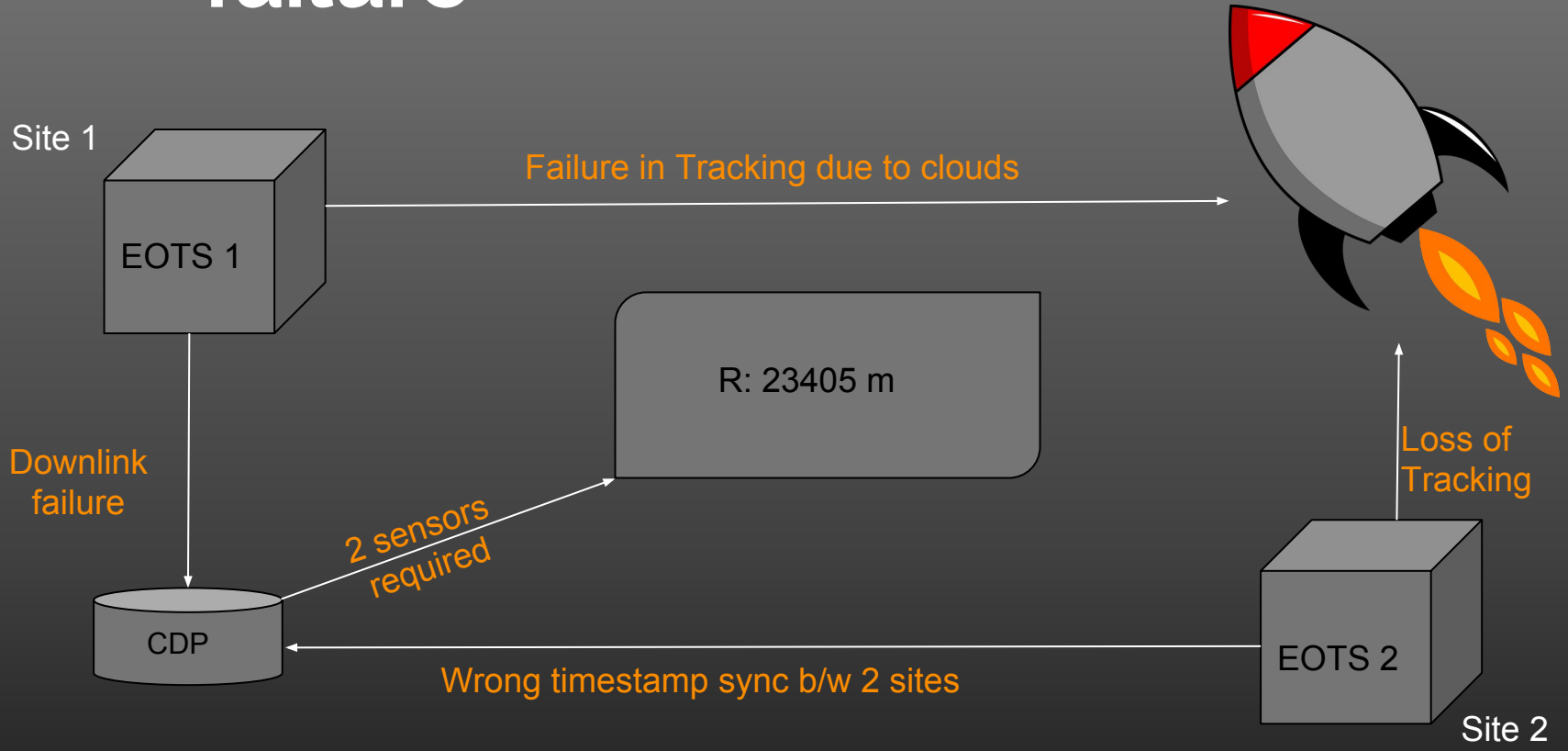
Defining EOTS (Electro Optical Tracking System)



Workflow


Site 1
EOTS 1
Tracking the object in FOV
R: 23405 m
DownLink
Geometric methods
Tracking
CDP
EOTS 2
Azimuth, Elevation, Timestamp to base
Site 2

Causes of failure



Methods at our disposal

1. Geometric calculation(Monocular Vision) - Deterministic method
2. Machine Learning
   a. Linear Regressor
   b. SVM regressor
   c. MLP regressor
3. Deep Learning(CNNs)

Monocular Vision

Digital (CMOS) Camera used for taking pictures from which depth map is estimated

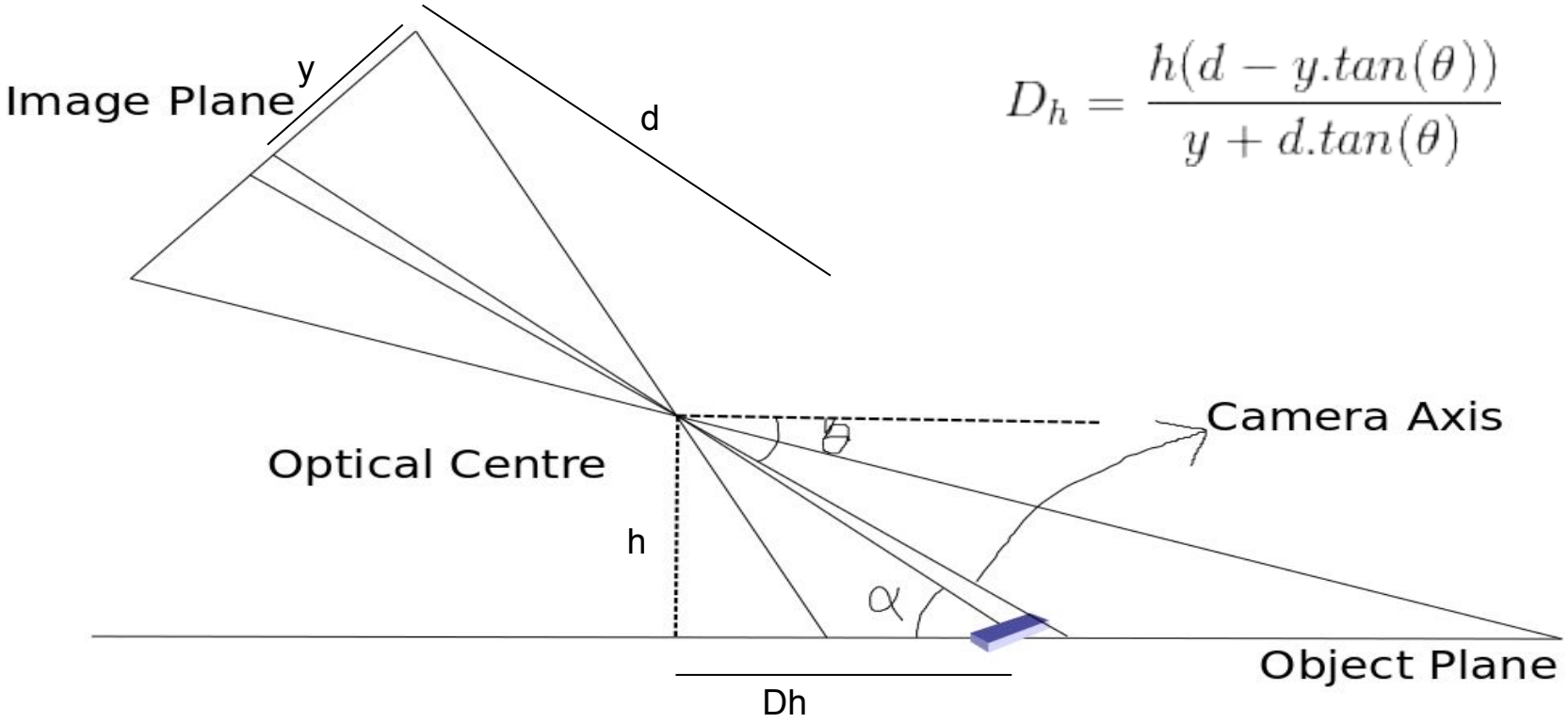
Some groundwork

Using geometrical relations and pinhole camera approximations we have,

1. Fixed camera pose, and position; i.e. we know the camera's position, height and angle of depression of the optical axis.
2. Also given the focal length of the lens of the camera, and resolution of the image taken,

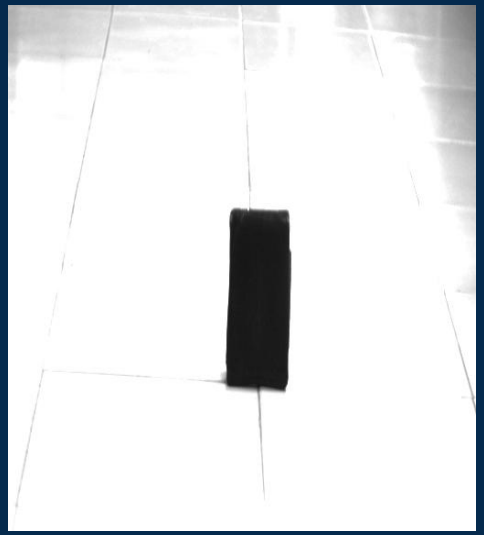
We can approximate the position of the object in cartesian coordinate. Codes have been written that return values in accuracy of 98% and above for ranges <2m.

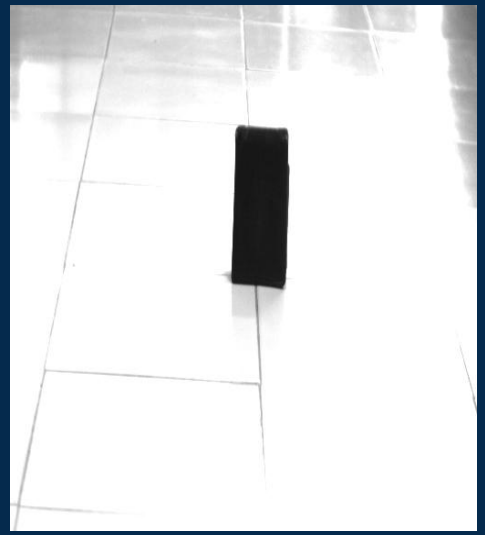
Arrangement of Apparatus for the Experiment



$$D_h = \frac{h(d - y.tan(\theta))}{y + d.tan(\theta)}$$

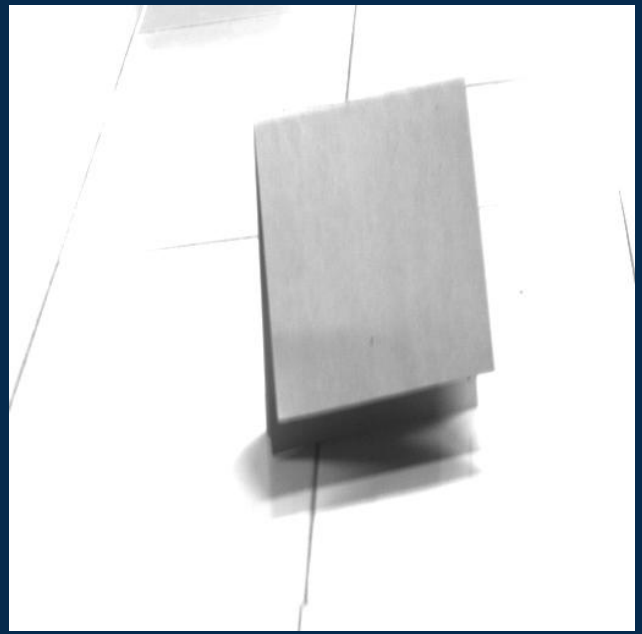
How did we do it?

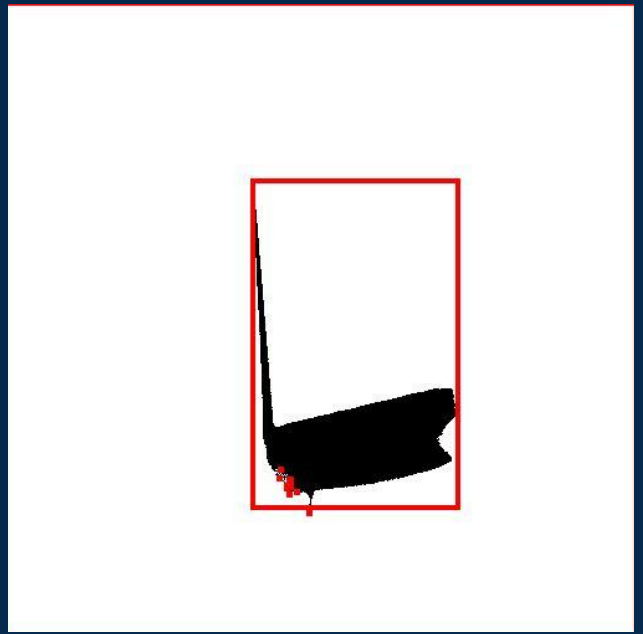




Take images with various ranges

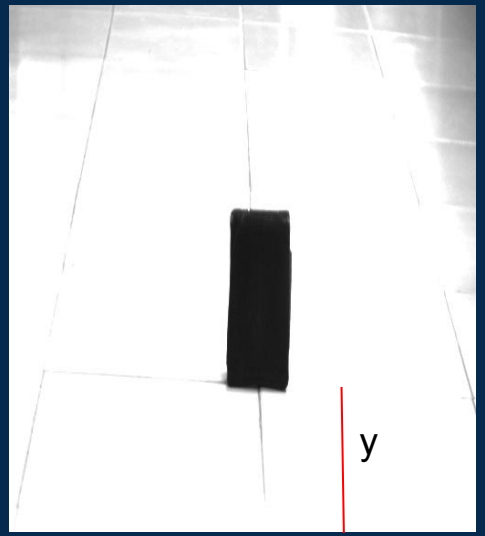
Segment the image: Thresholding (OTSU)





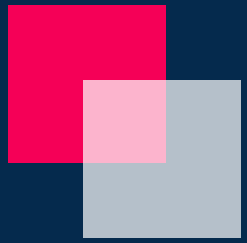
An example

1. Threshold the image - Segmentation
2. Compute the centroid or the centre of an edge, in this case centre of the bottom edge
3. Calculate y from the image
4. With other given parameters in the previous equation estimate the range Dh.



Key results

Actual Range	Predicted Range
1.59 m	1.581 m
1.46 m	1.432 m
1.40 m	1.396 m
2.00 m	1.894 m
4.50 m	3.362 m



Problems

1. Only applicable in 2D plane.
2. The accuracy suffers a lot of damage for range>3m
3. The results depend a lot on parameters that can be hidden, i.e. shape and size.
4. Error diverges when the value of R increases.

Statistical Methods

Where to begin?

Prithvi = Dataset!

Dataset

Creating a new Dataset

- **From?**
  Tracked videos with annotations (EOTS output).
- **How?**
  - Extract range - ROI based OCR
  - Filter range data
  - Make a dataset
- **Why?**
  - No previous data available

Construction of the Dataset

1. Given a video as an input, extract frames.
2. Construct an array of structure, where each structure contains a frame and its corresponding range (extracted from annotation) in the frame as elements.
3. Now separate the elements into arrays and filter the range which is unacceptable.
4. Also, the values which are ill poised are corrected using a filter designed by DRDO
5. The dataset is now formed by separating them into two arrays of Images and Range

Practical insight

An annotated video is the input, where we extract each frame as a datapoint.

Practical insight

From each frame the Range is Extracted from the annotation at the bottom right corner.

Problems encountered:

- Bad stamping - Line Errors
- Intensity values of font and background were same
- ROI might change
- 0 , 8 are almost matched as the same

Practical insight

Now that portion or Region Of Interest (ROI) is segmented.

Segmentation technique:-

- OTSU's technique FAILS!
- Sample the digit pixel value
- Use Region Growing to enhance
- Threshold the ROI

Practical insight

The target for training is in this ROI and hence we need to convert this into meaningful range.

How do we separate into digits:

- Keep summing over y, moving along x
- Where the sum over y for given x is 0, is end of an digit
- Separate and start looking till the sum is positive.
- Again segregate once sum is 0
- Template making

Practical insight

Use Template matching as,

1. Always same font is used
2. At the same scale
3. At the same position

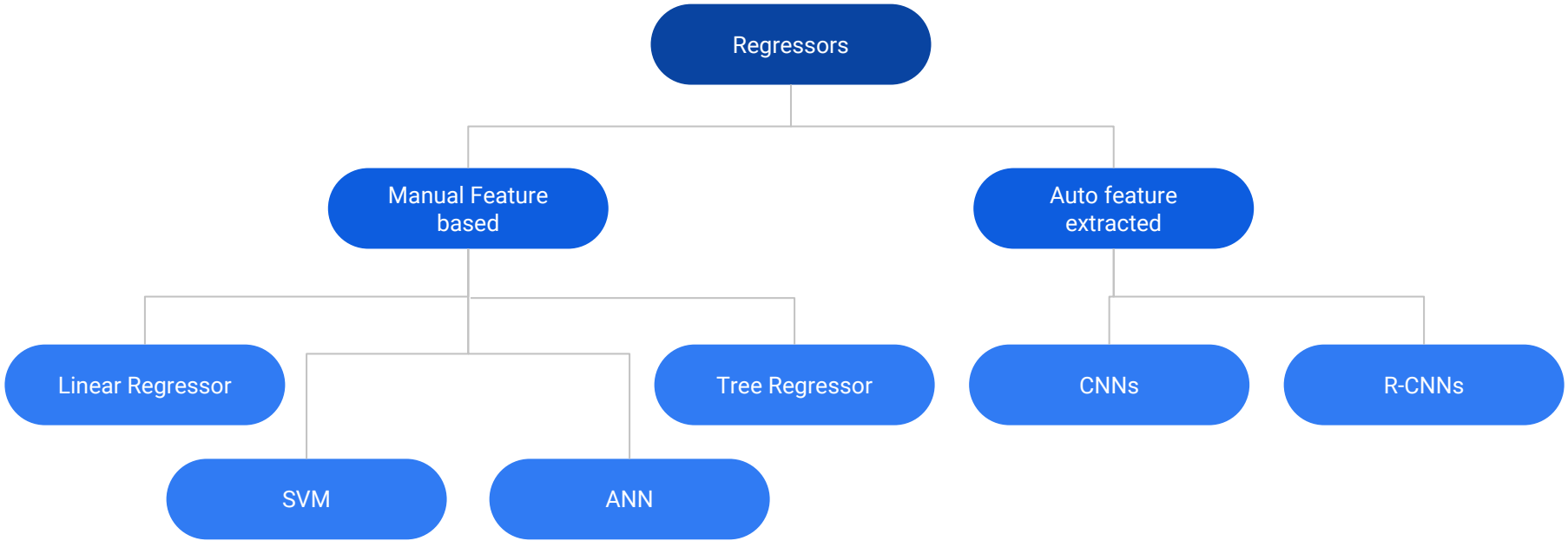
All these favour the above

D : 0132570

Choose Machine Learning or Deep Learning?

Use both, compare and find the better fit!

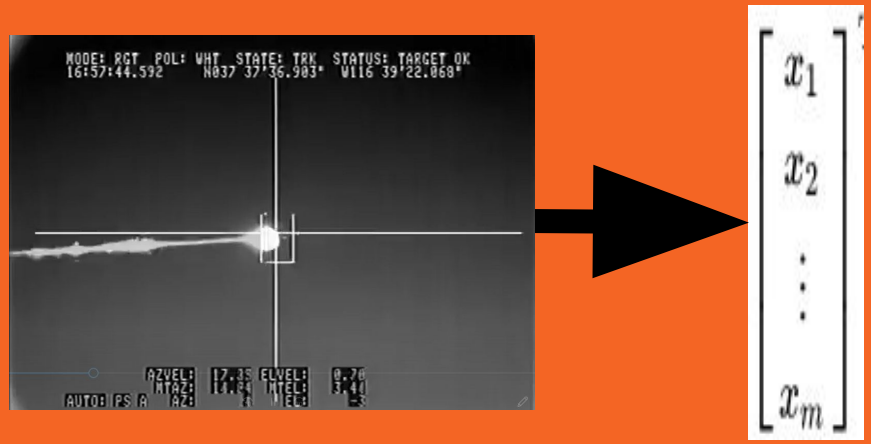
Approach


Regressors
Manual Feature based
Auto feature extracted
Linear Regressor
SVM
ANN
Tree Regressor
CNNs
R-CNNs

Practical insight

Regressor Design

We have the training data, and also targets we need to transform the images to feature vector, i.e. we need to manually extract features.

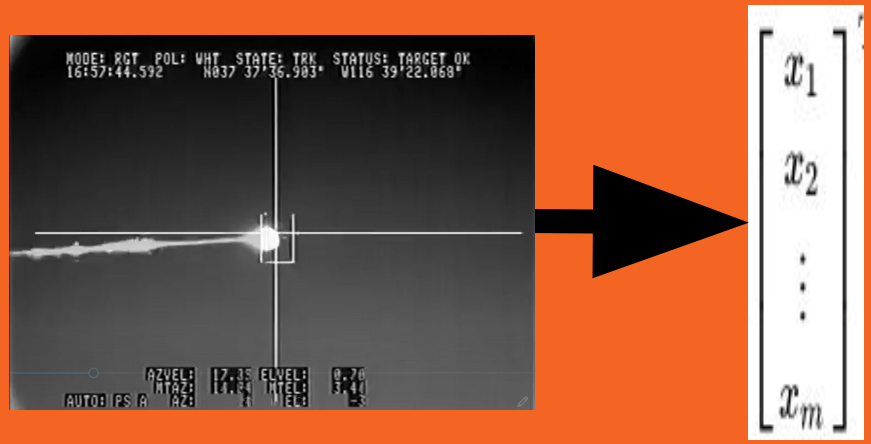


Practical insight

Feature Extraction

What are the features we have used:

1. Size of the object
2. Average Intensity of the object
3. Relative Intensity of the object
4. Eccentricity of the object

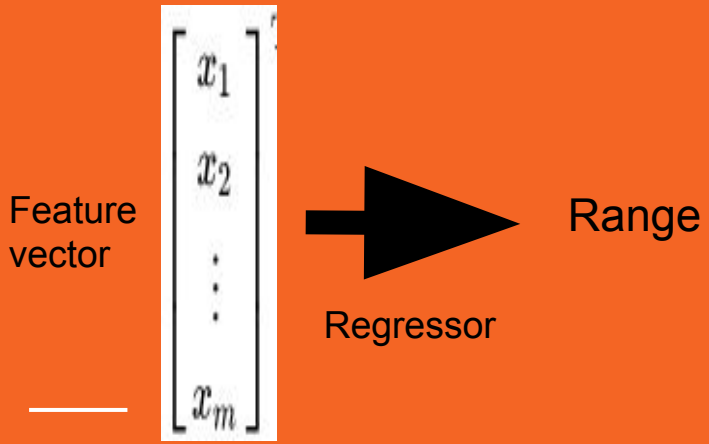


$$R = f(F_v)$$

Practical insight

Regressor design

‘f’ is the regressor,
‘Fv’ is the Feature Vector
We need to find an optimal ‘f’



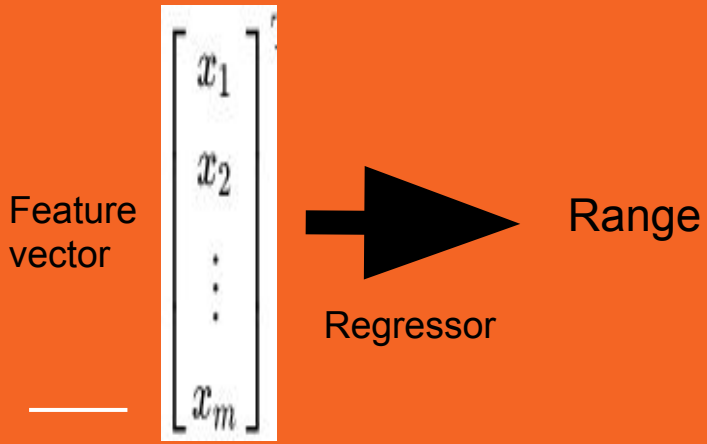
$$R = w^T F_v$$

Practical insight

Linear Regression

'w' is the weight matrix,
'Fv' is the Feature Vector
We need to find an optimal 'w'

RMSE = 25.086 km

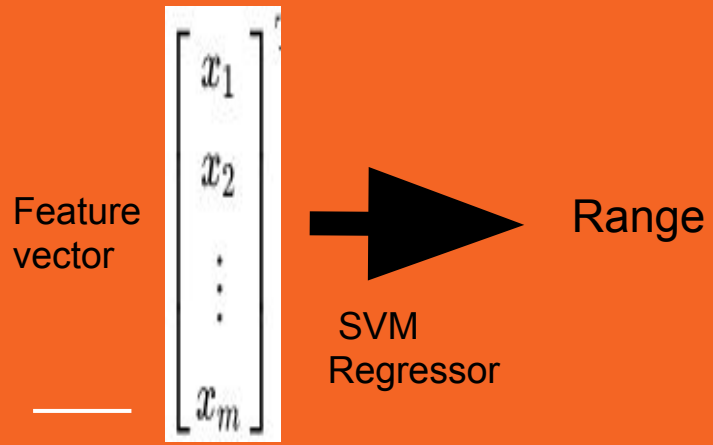


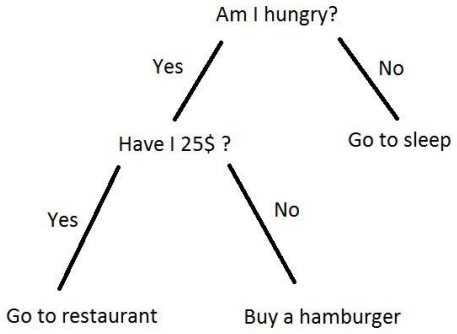
Practical insight

SVM Regression

A generic algorithm has been used

RMSE = 23.108 km linear
RMSE = 21.186 km nonlinear



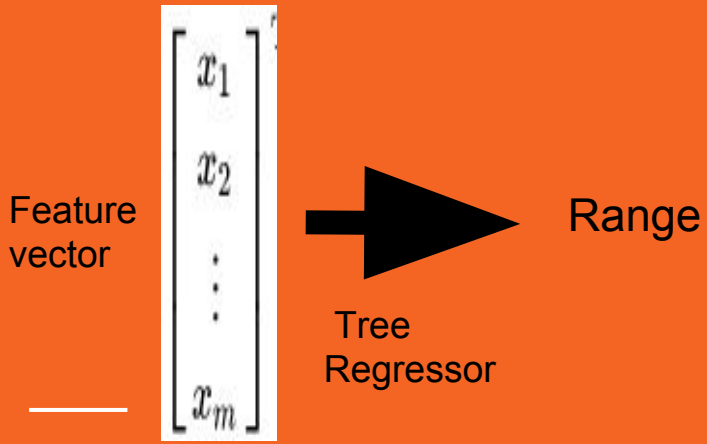


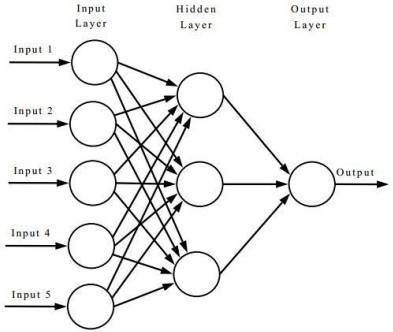
Practical insight

Decision Tree Regression

RMSE = 4.1946 km

Divides the problem space into regions in which the data is linearly separable, and implements a linear classifier in each interval.





Practical insight

ANN Regression

RMSE = 9.3901 - 10 Neurons

RMSE = 8.0328 - 15 Neurons

RMSE = 7.0915 - 20 Neurons

RMSE = 8.3112 - 100 Neurons

Feature vector

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

ANN Regressor

Range

CNNs for Regression

- CNNs eliminate the need for manual feature extraction—the features are learned directly by the CNN.
- CNNs have been known to produce state-of-the-art recognition results.
- CNNs can be retrained for new regression tasks, enabling you to build on pre-existing networks.

Is it still worth the complexity? Let's find out

Training Data | Validation Data | Test Data

4760/6800 samples in the dataset are assigned as the training data, on which the network/regressor - 70 % split(60 % training - 10 % validation)

2040/6800 samples are used as test data, for validation purposes.

Special care is taken that the data is uniformly distributed in both the cases.

Architecture

Neural Network's structure

- **Deep Learning?**
  Need to see results to see if it is required
- **How?**
  - Design each layer
  - Compatibility of adjacent layers
  - Decide how deep
- **Why?**
  - Parameters affect accuracy

Hyperparameters

Various parameters to play with

- **Network Architecture**
  Number of layers, Arrangement of layers
- **Learning Rate**
  How fast the weights change to converge
- **Regularization**
  Dropouts help network avoid overfitting

Constraint one:
Ratio of real filter size to receptive field size is greater than ⅙.

Information

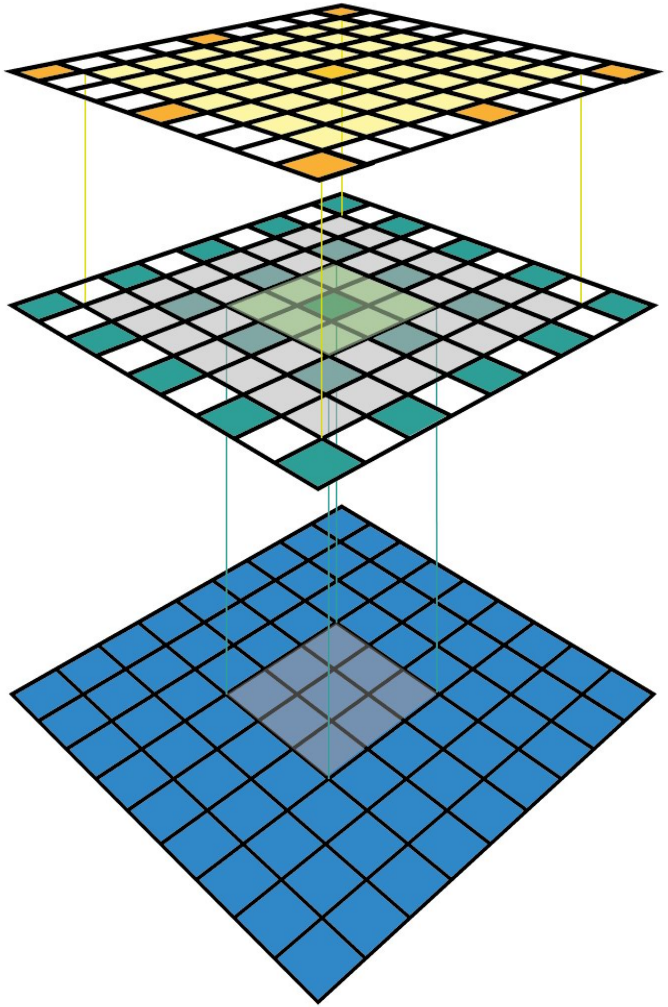
The **receptive field** is defined as the region in the input space that a particular CNN's feature is looking at (i.e. be affected by).

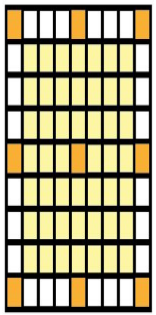
Another one!

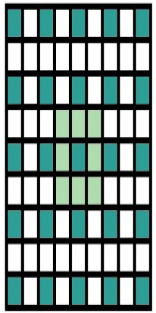
The receptive field size of the last neuron should not be the whole img.

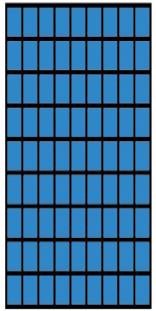
Fact

Each successive layer in CNN is capable of learning increasingly abstract features of the original image due to its unique features in receptive field.









Network - I

Image Input Layer(152x152)

Convolution Layer(3x3, 1)

ReLU Layer

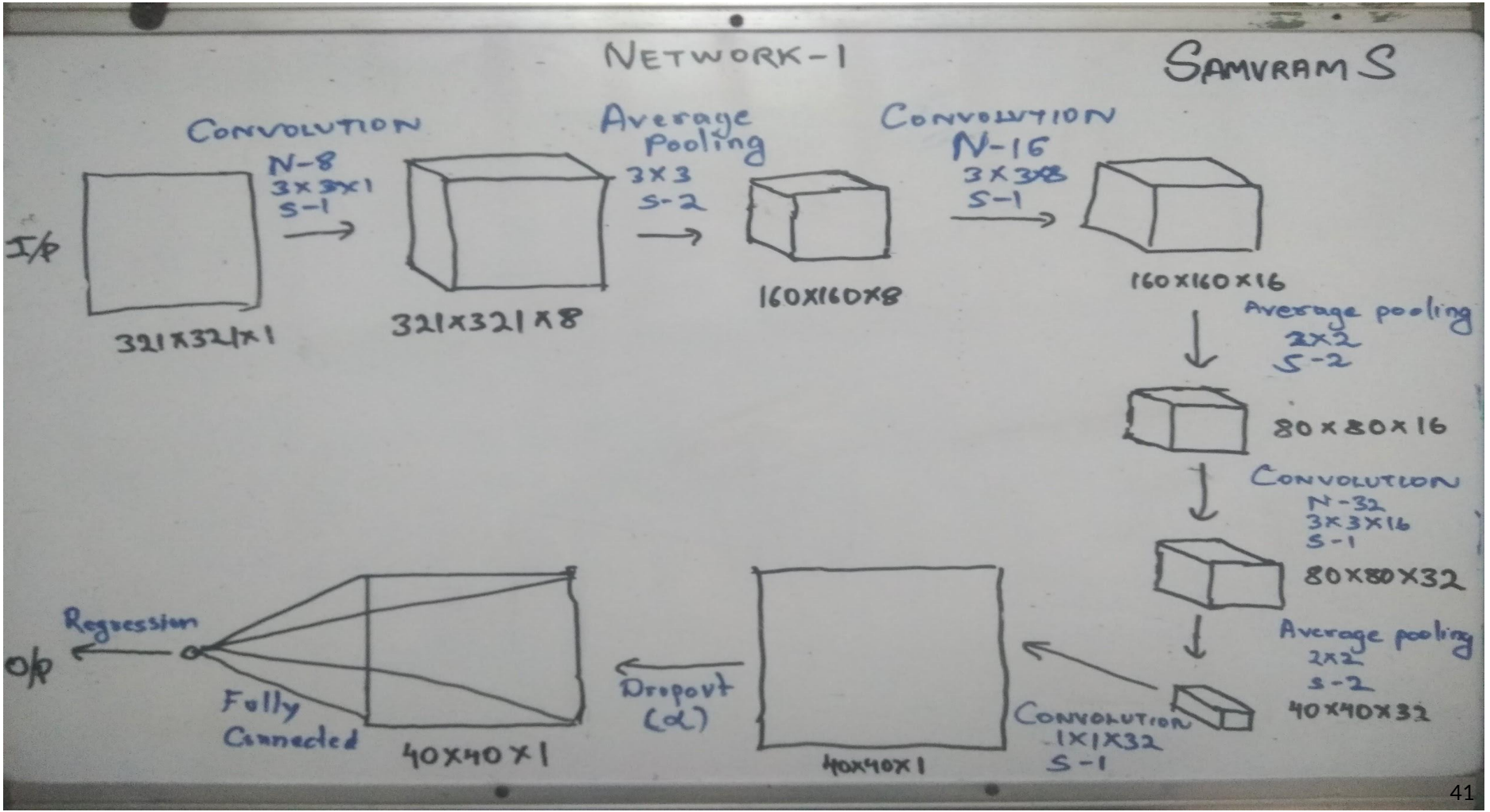
Normalization Layer

Average Pooling Layer

3

Fully Connected Layer

Regression Layer


NETWORK-1
SAMVRAM S
CONVOLUTION
N-8
3X3X1
S-1
I/p
321X321X1
321X321X8
Average Pooling
3X3
S-2
160X160X8
CONVOLUTION
N-16
3X3X8
S-1
160X160X16
Average pooling
2X2
S-2
80X80X16
CONVOLUTION
N-32
3X3X16
S-1
80X80X32
Average pooling
2X2
S-2
40X40X32
CONVOLUTION
1X1X32
S-1
40X40X1
Dropout
(α)
40X40X1
Fully
Connected
Regression
O/p

Network - 1

A crude architecture made for making it a starting point.

1. Combination of Convolution, Normalization and ReLU layer with avg pooling layer is a stable architecture backbone.

2. Depth of the network can be decided on the previous 2 constraints, and here intuitively taken as 3.

Network - 1

A crude architecture made for making it a starting point.

1. The DropOut Layer is a regularization parameter that curbs overfitting.

2. Optimize α for this network

Network - II

Image Input Layer(152x152)

Convolution Layer(3x3, 1)

ReLU Layer

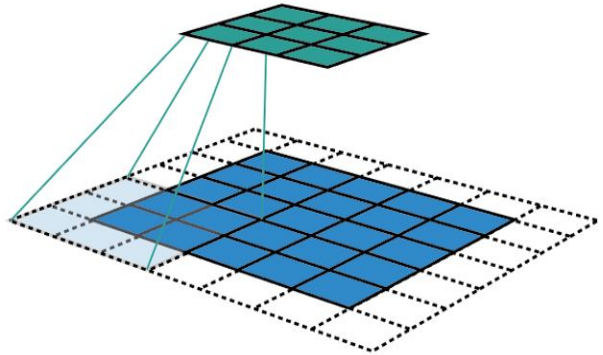
Normalization Layer

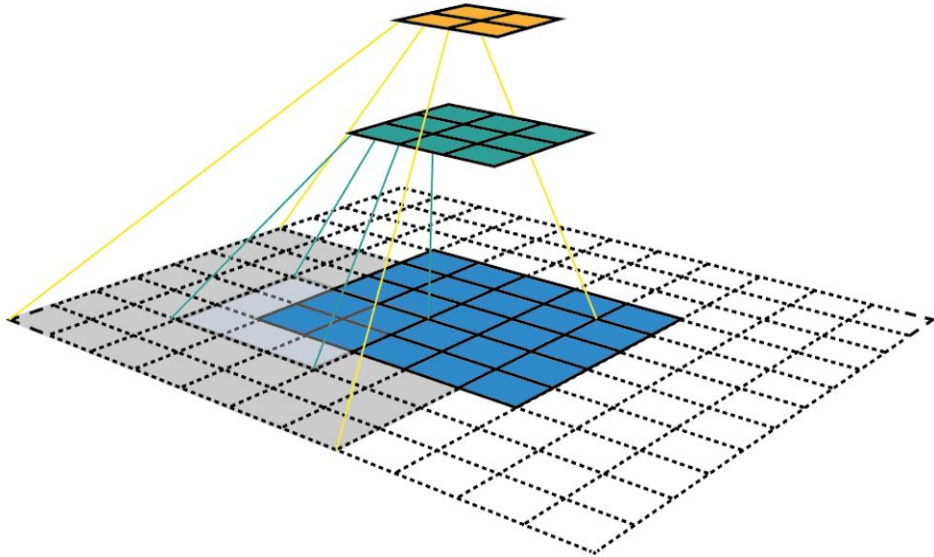
Max Pooling Layer

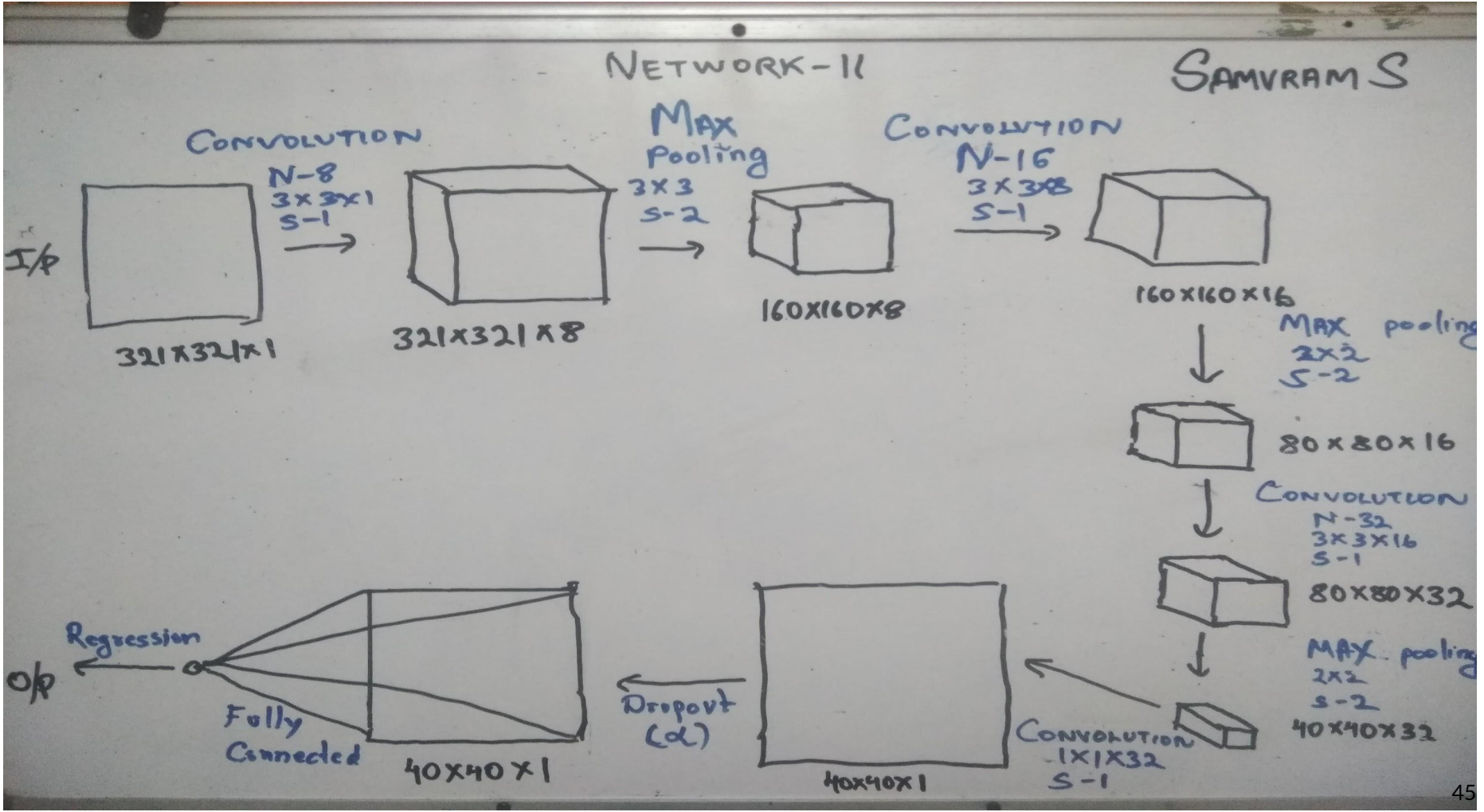
3

Fully Connected Layer

Regression Layer





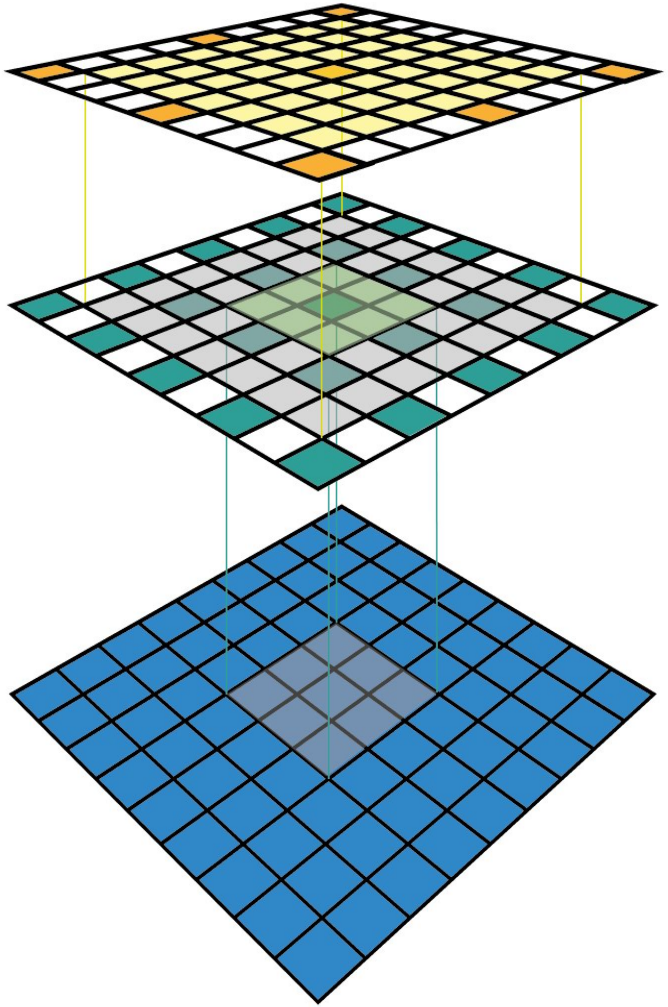

NETWORK-11
SAMVRAM S
CONVOLUTION
N-8
3X3X1
S-1
I/p
321X321X1
321X321X8
MAX
Pooling
3X3
S-2
160X160X8
CONVOLUTION
N-16
3X3X8
S-1
160X160X16
MAX pooling
2X2
S-2
80X80X16
CONVOLUTION
N-32
3X3X16
S-1
80X80X32
MAX pooling
2X2
S-2
40X40X32
CONVOLUTION
1X1X32
S-1
40X40X1
Dropout
(α)
40X40X1
Fully
Connected
Regression
O/p

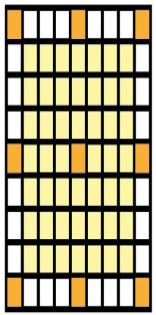
Network - 2

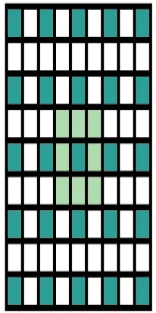
A contrast to network 1, by replacing average pooling with max pooling.

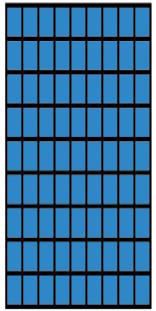
1. On comparing the results of Network-1 and Network 2 we can determine which pooling is better suited for our use.

2. Ɗ is also needed to be tuned for this network and is expected not to vary much.









Network - III

Image Input Layer(152x152)

Convolution Layer(3x3, 1)

ReLU Layer

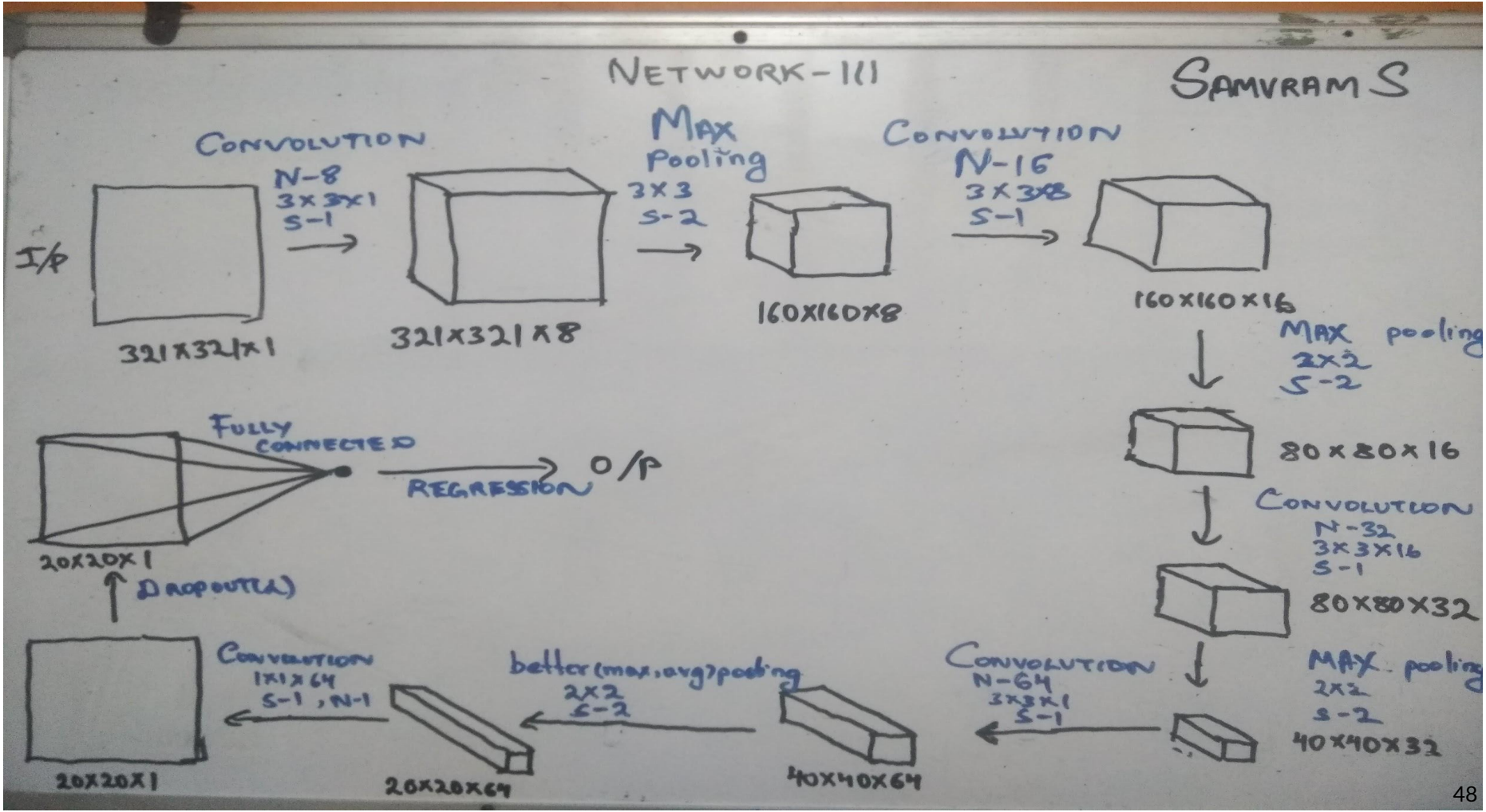
Normalization Layer

Better of Avg/Max Pooling Layer

4

Fully Connected Layer

Regression Layer


NETWORK-III
SAMVRAM S
CONVOLUTION
N-8
3X3X1
S-1
I/P
321X321X1
MAX
Pooling
3X3
S-2
321X321X8
160X160X8
CONVOLUTION
N-16
3X3X8
S-1
160X160X16
MAX pooling
2X2
S-2
80X80X16
CONVOLUTION
N-32
3X3X16
S-1
80X80X32
MAX pooling
2X2
S-2
40X40X32
CONVOLUTION
N-64
3X3X1
S-1
40X40X64
better (max, avg) pooling
2X2
S-2
20X20X64
CONVOLUTION
1X1X64
S-1, N-1
20X20X1
DROPOUT(.4)
20X20X1
FULLY CONNECTED
REGRESSION
O/P

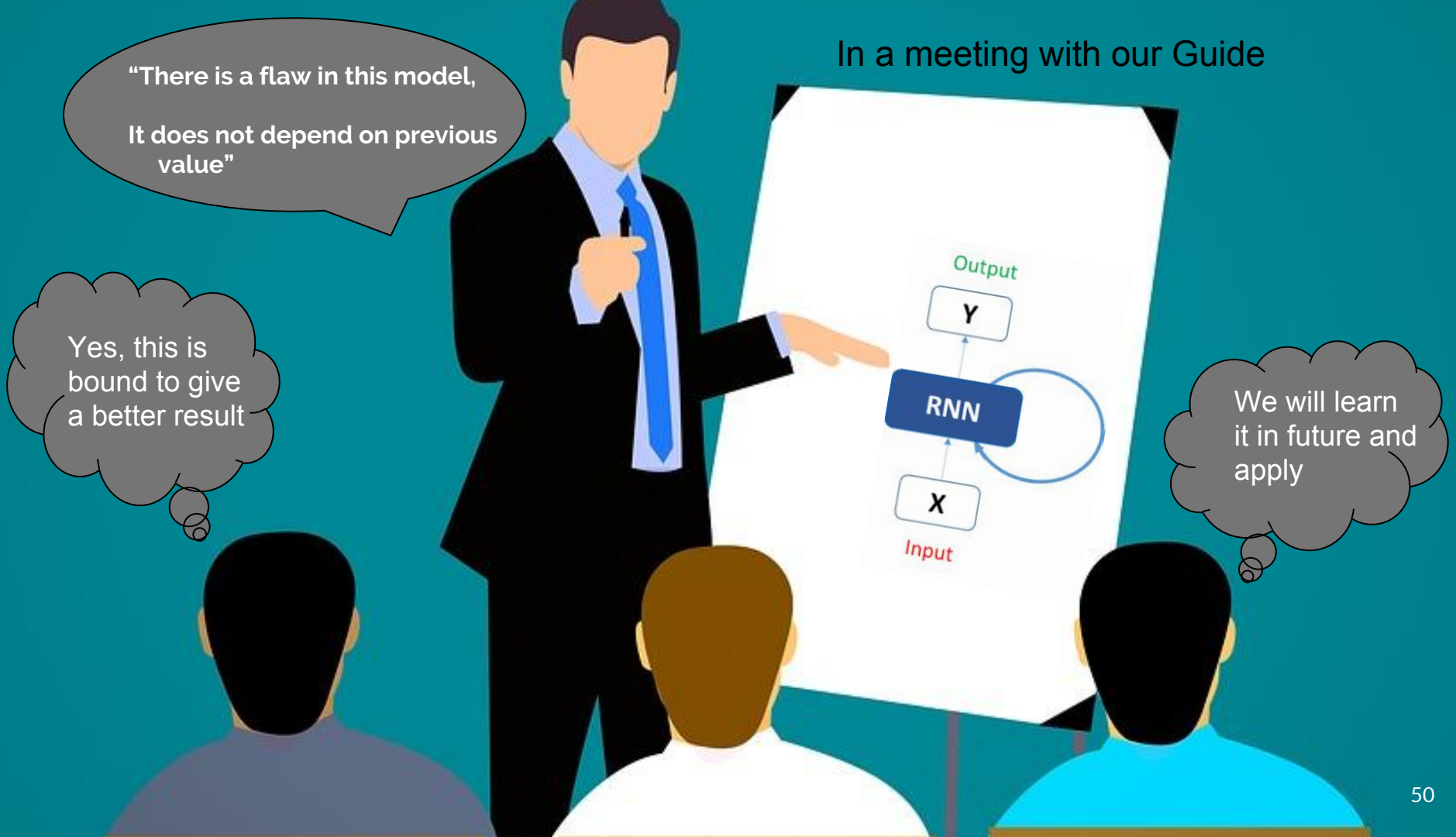
Network - 3

A contrast to network 1 & 2, to see the results of increasing depth of the CNN.

1. On comparing the results of Network-1,2 and Network 3 we can determine if the depth can be increased further without affecting the accuracy.

2. Ɗ is also needed to be tuned for this network and is expected not to vary much.

In a meeting with our Guide



Network - IV

Image Input Layer(152x152)

Convolution Layer(3x3, 1)

ReLU Layer

Normalization Layer

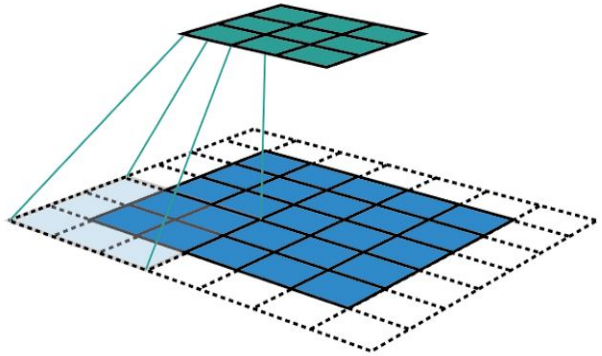
Max Pooling Layer

3

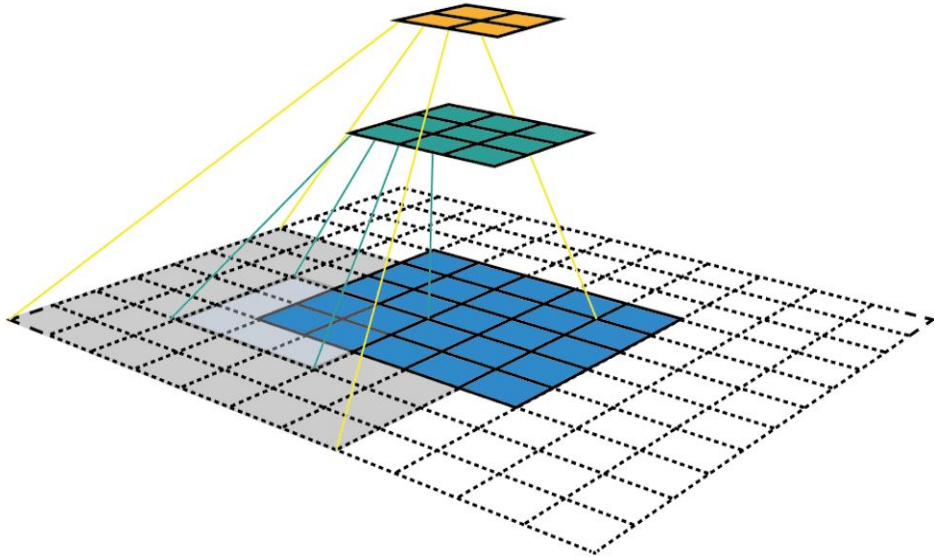
Fully Connected Layer

RNN/GRU Layer

Regression Layer



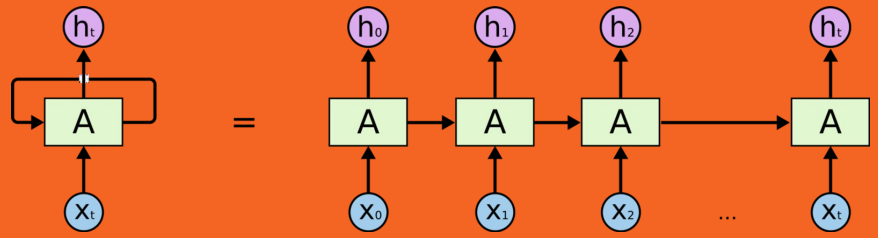
NOT IMPLEMENTED!

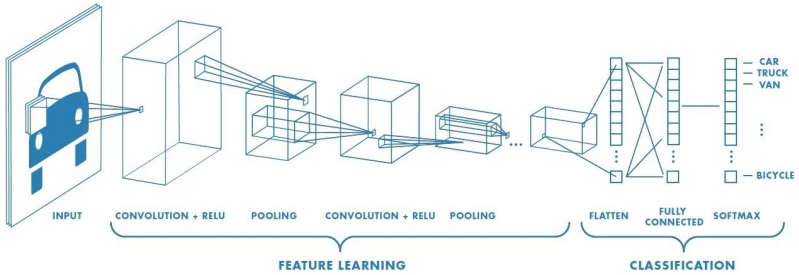


Highly experimental

A go at R-CNN for using the memory of previous predictions to correct and better the readings

A **recurrent neural network** (**RNN**) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit temporal dynamic behavior for a time sequence.





Practical insight

CNN for Regression

RMSE = 15.636 km

After tuning, and making some changes to few training loops,

RMSE = 10.386 km

Range

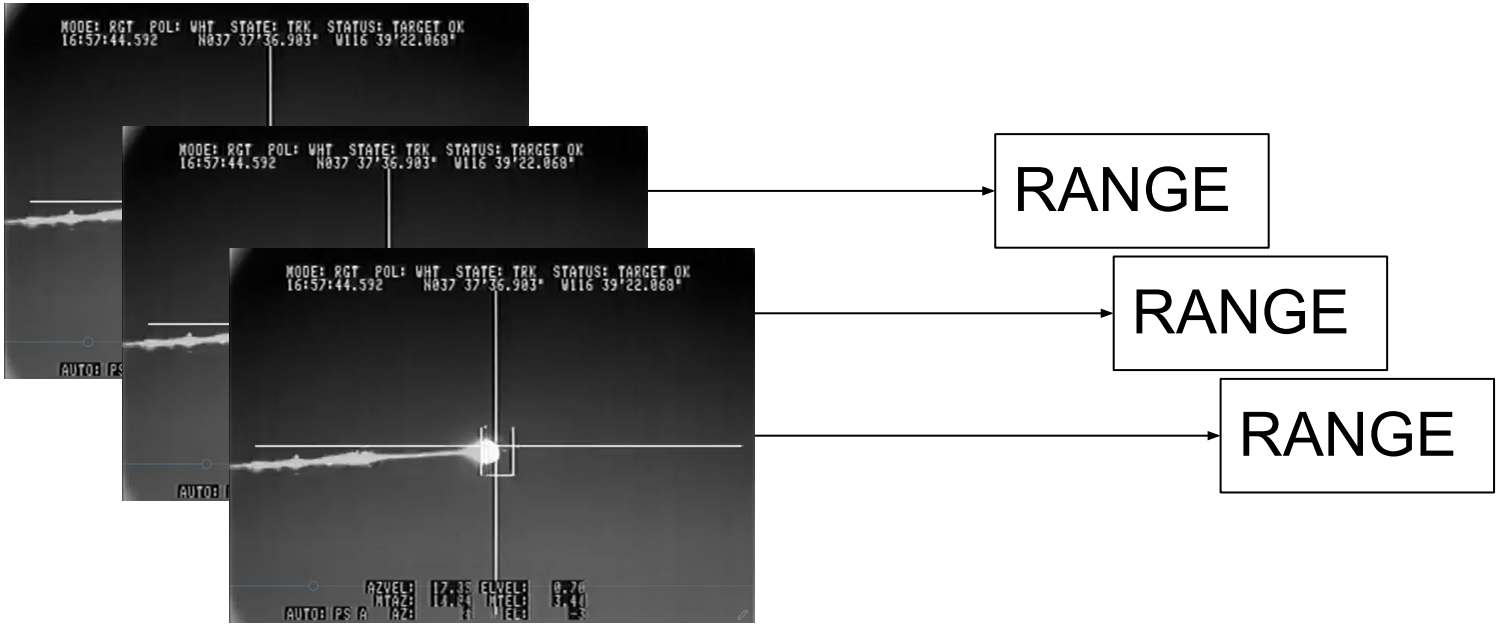
CNN Regressor

RESULTS

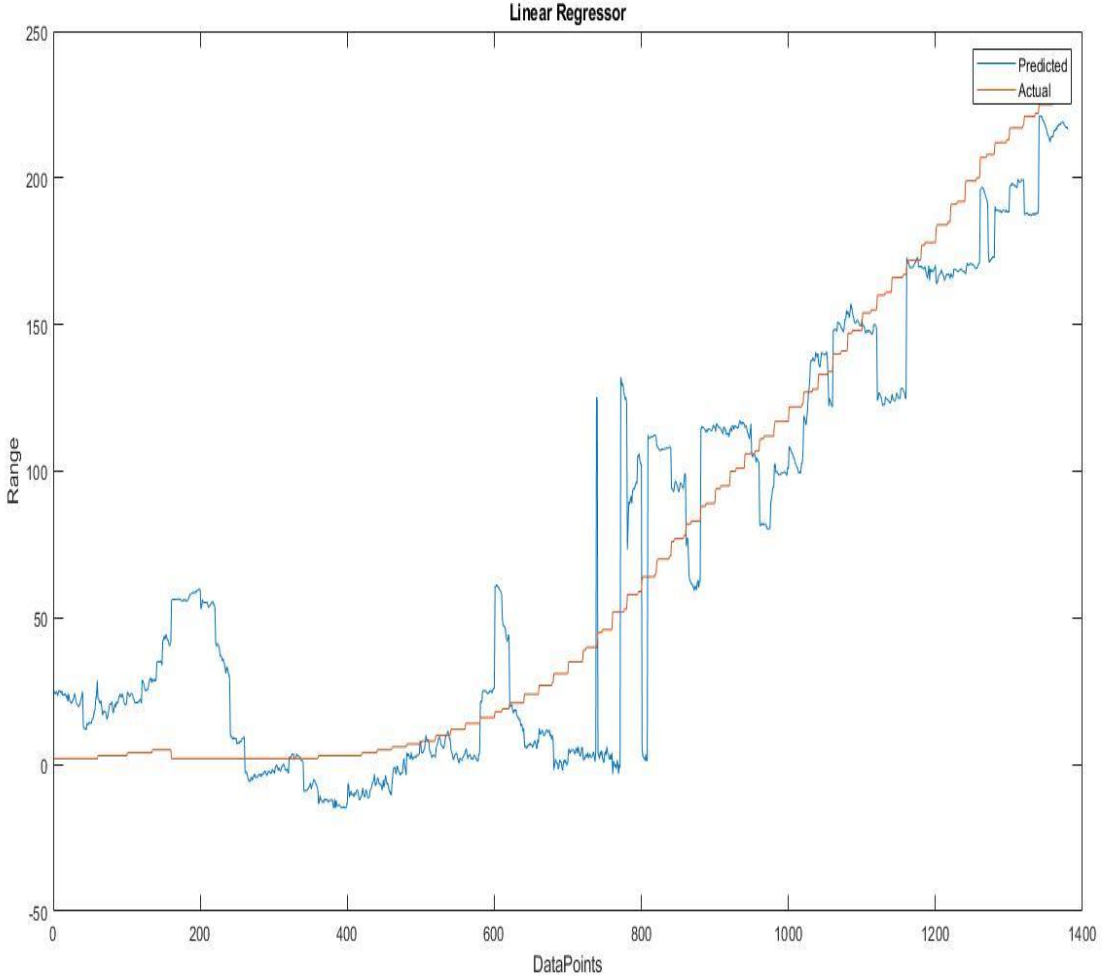
&

INFERENCES

Assumption : Each frame when analysed provides a value of range **independent** of the other frames

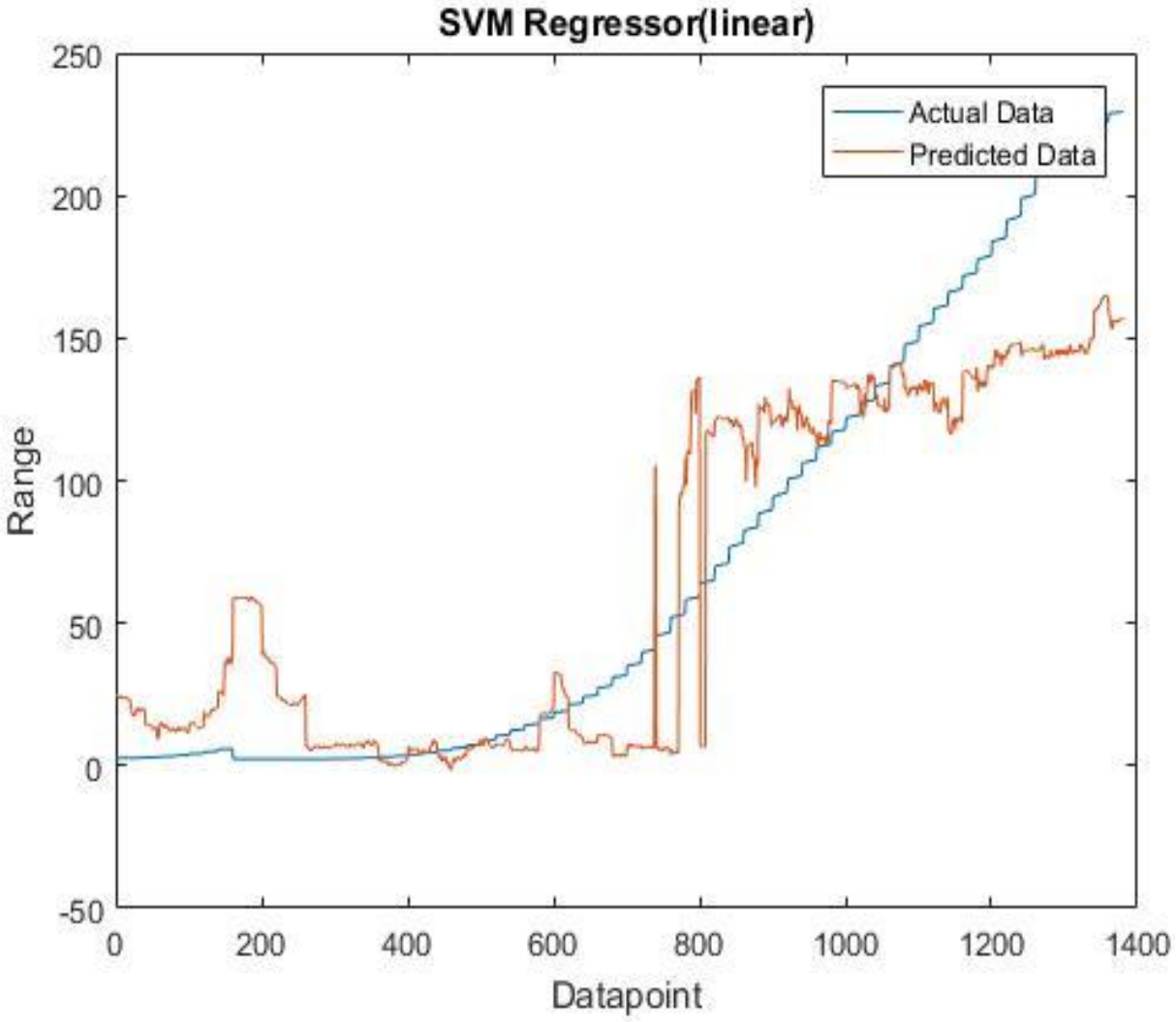


This assumption is not true, as the range has a dependency on the previous frame value, which shows RNNs would be a better fit at solving the above problem.



Inference - The map from the feature space to Range has some complexities (non-linearities) which a linear regressor fails to capture

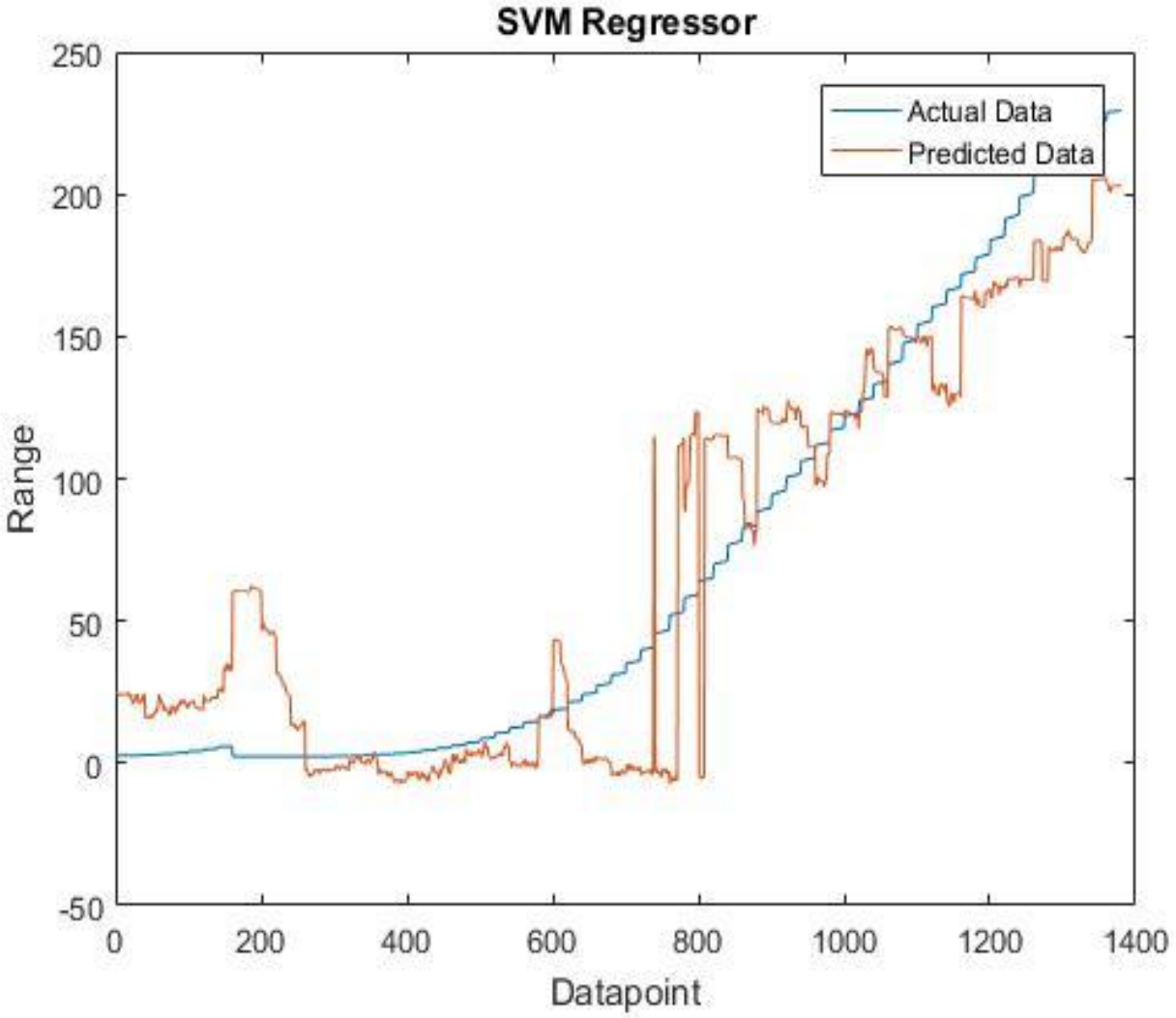
Hence we move on to SVM class of regressors.



Inference -

SVMs do not perform well on highly skewed/imbalanced data sets. These are training data sets in which the number of samples that fall in one of the classes far outnumber those that are a member of the other class. Customer churn data sets are typically in this group because when you collect the training set, among a million customers during a particular time period, there would be very few who have actually churned .

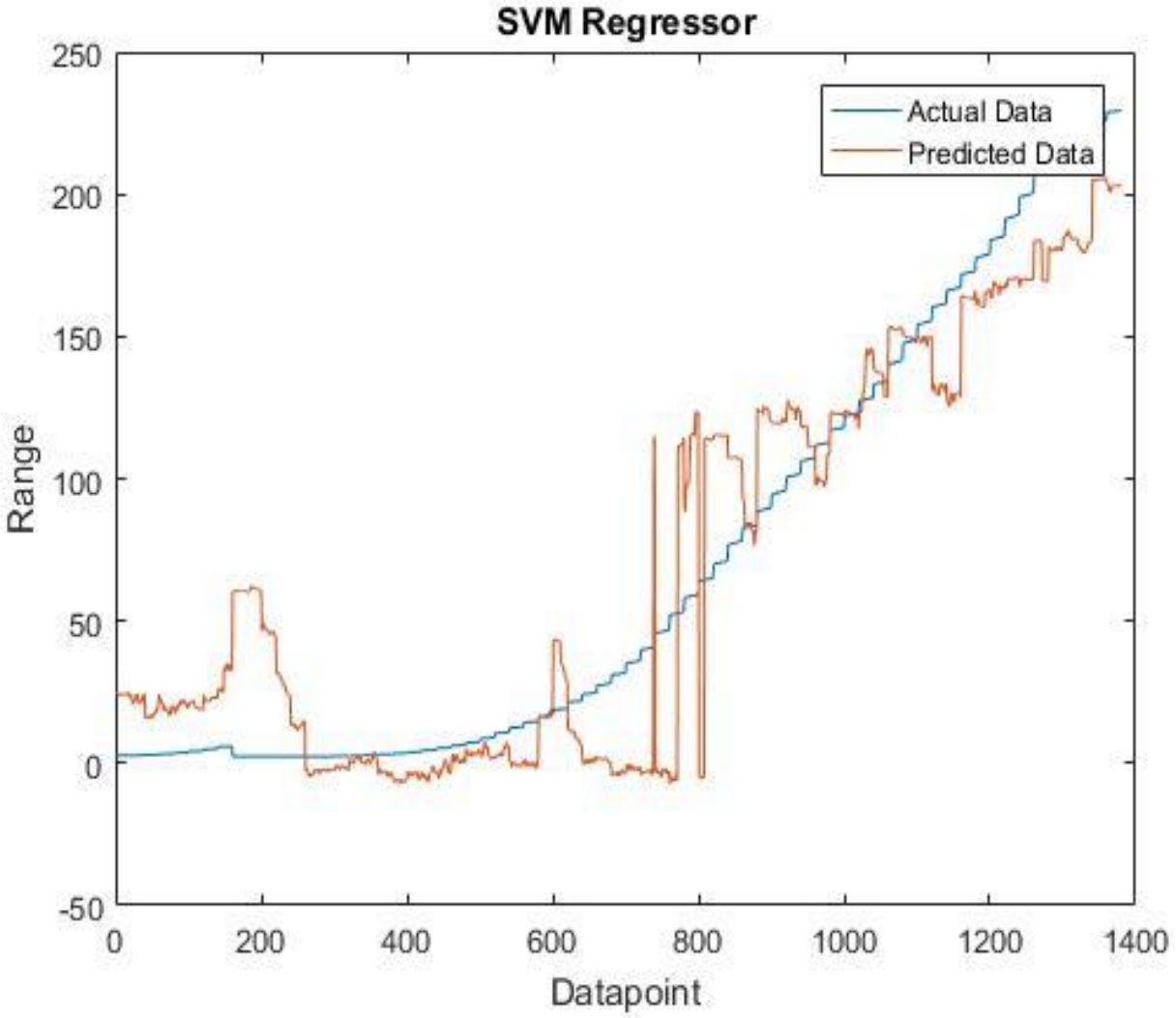
SVMs are also not a good option especially if you have multiple classes. Ultimately in this case, you get back to a binary classifier and then use some kind of a voting mechanism to classify a sample to one of the classes.



Inference -

Non-linear SVMs perform better proving a complex non-linearity exists in the system. This in line with the previous observation we had.

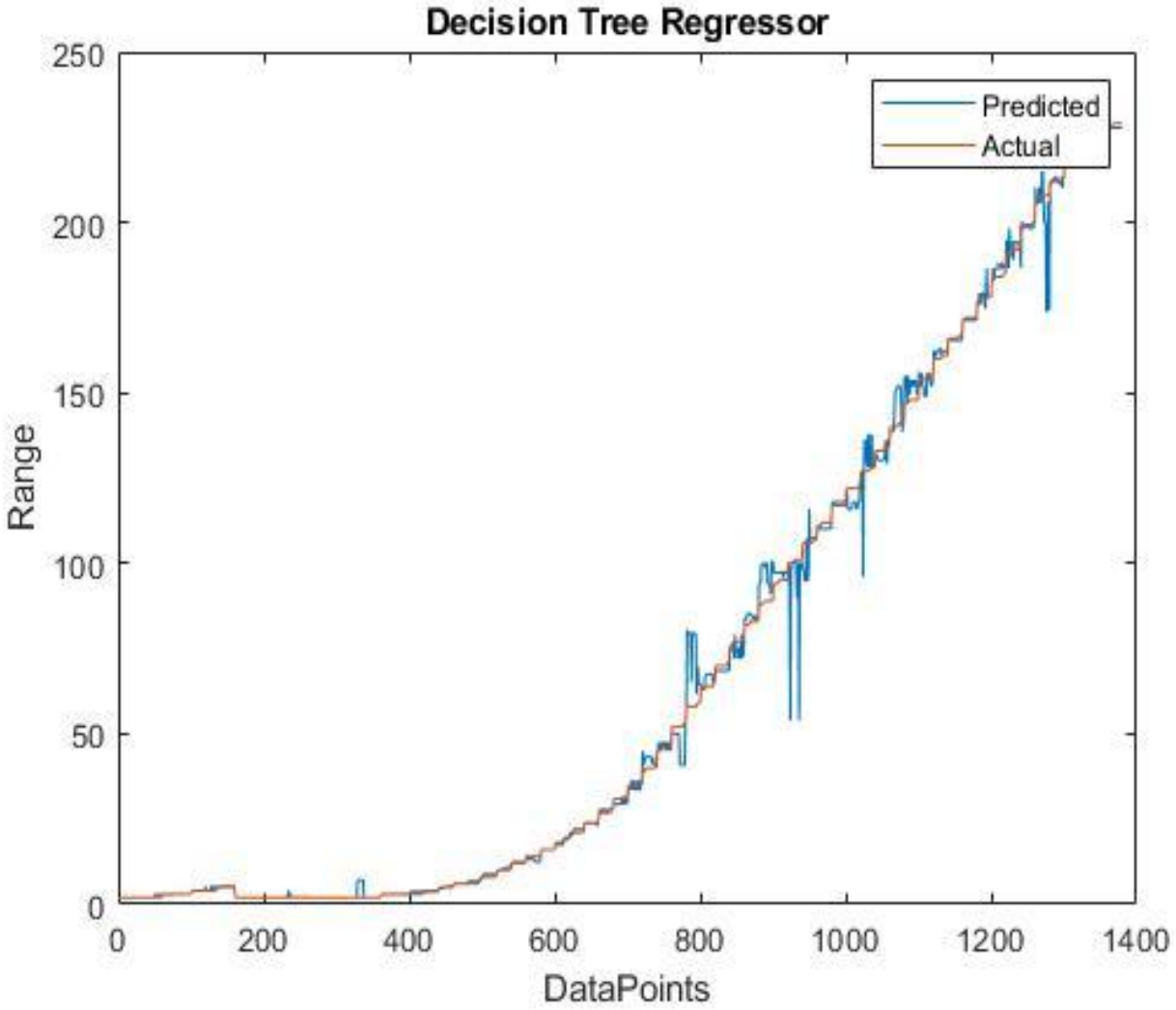
Hence we proceed to capture complex nonlinearities using ANN and Tree Regressor.



Inference -

Non-linear SVMs perform better proving a complex non-linearity exists in the system. This in line with the previous observation we had.

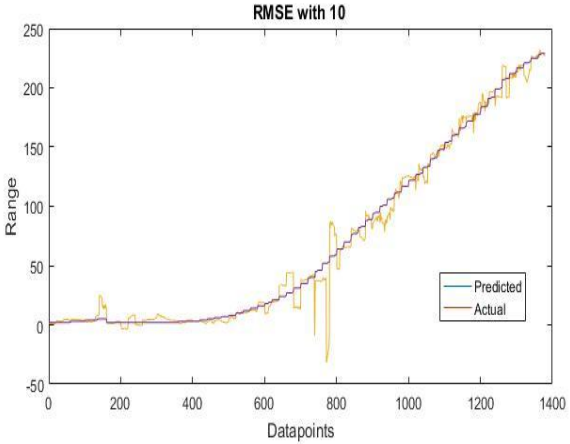
Hence we proceed to capture complex nonlinearities using ANN and Tree Regressor.

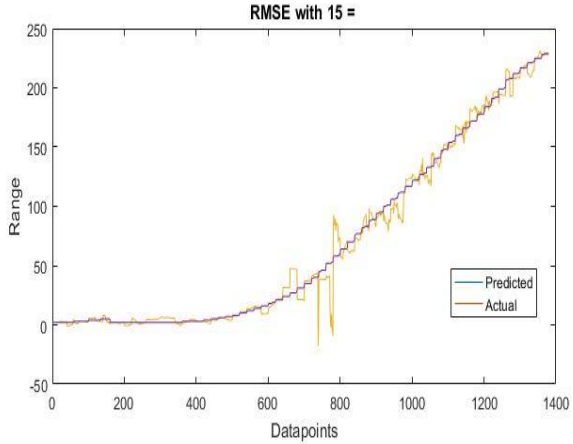


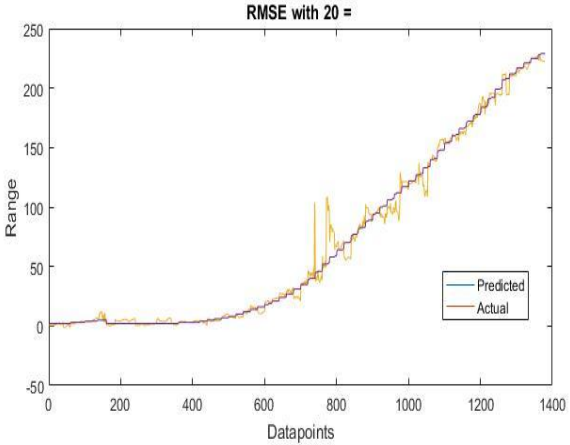
Inference -

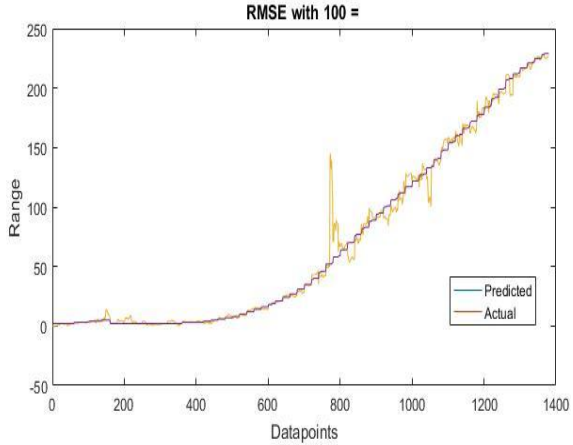
Decision tree regressor has produced best fits till now only because the training dataset and test dataset are more or less from the same video.

The scalability of this architecture is a big issue, hence we move to ANN regressor.





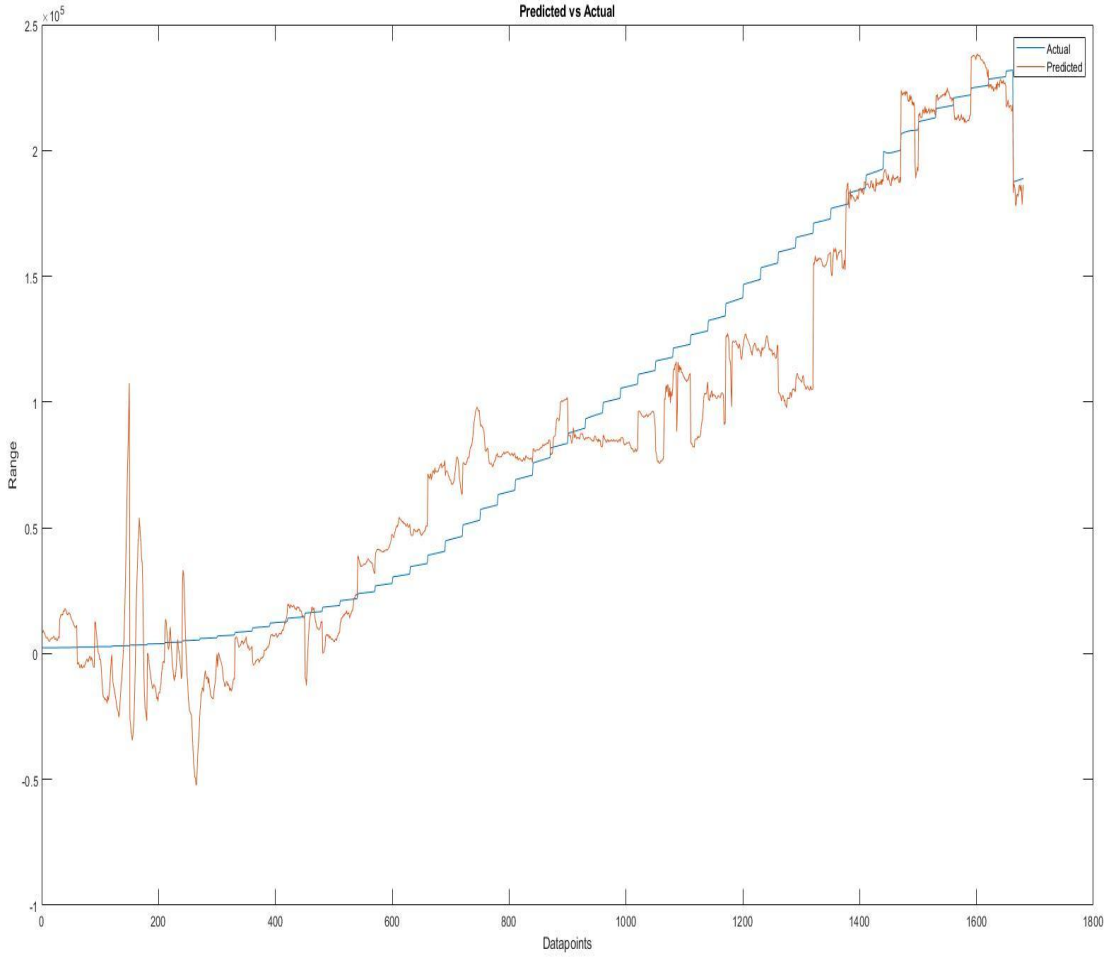




Inference -

ANNs are tuned by trial and error and are capable of capturing very complex non-linearities.

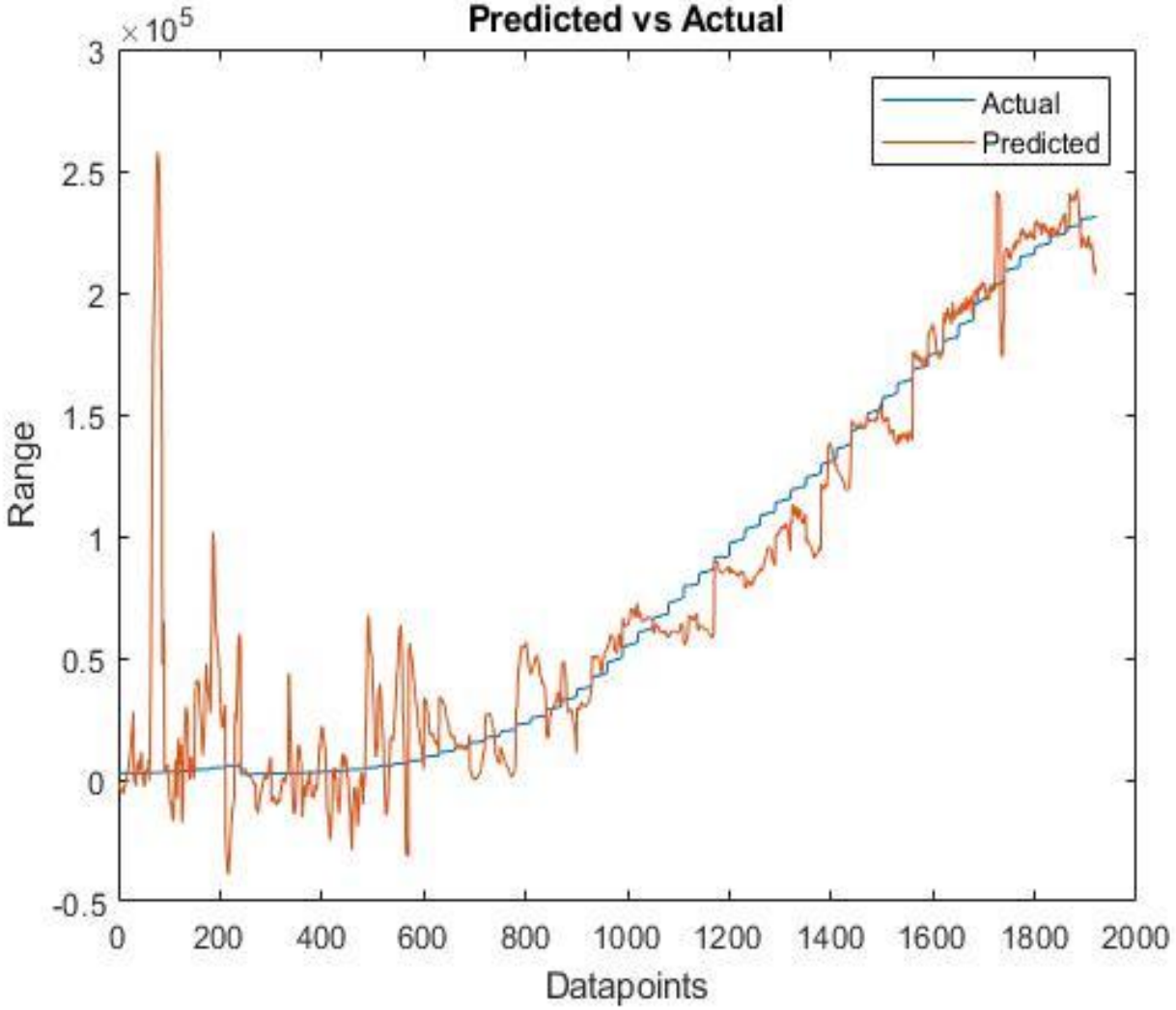
The results suggest a network architecture with 20 neurons in the hidden layer seems to outperform the other architectures.



Inference -

CNNs are best known for their generalization power when compared to other models. Incidentally, they also deliver state of the art results for recognition/classification tasks.

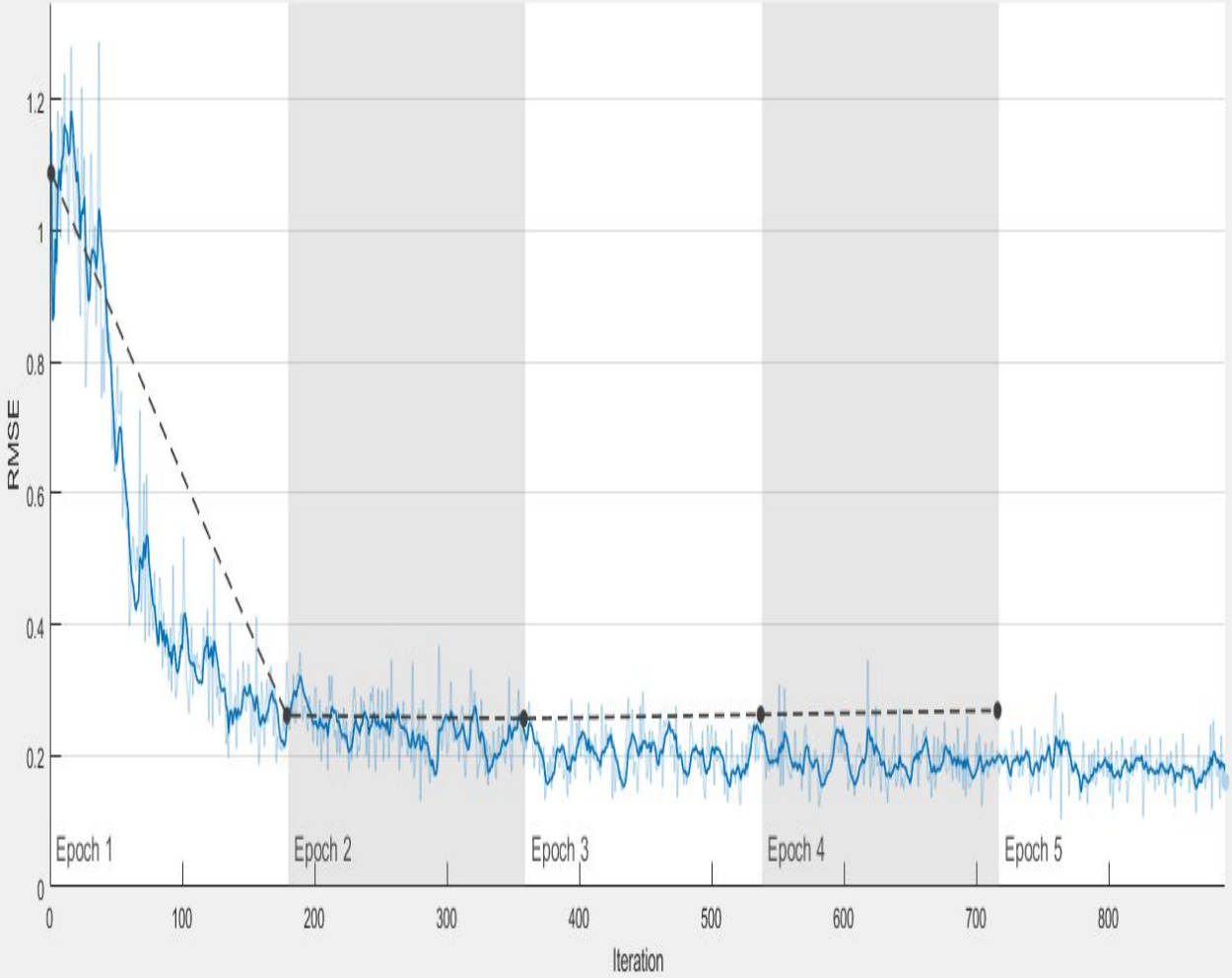
However the network is supposedly delivering generalisation at the cost of accuracy in our case.



Inference -

When complexity of CNN is reduced, i.e. a layer from the architecture is removed, we see the CNN performs well on larger ranges and performs very poorly on the smaller ranges.

This seems natural, as at the initial stages, active propulsion systems emitt plumes which work as noise, for CNN.

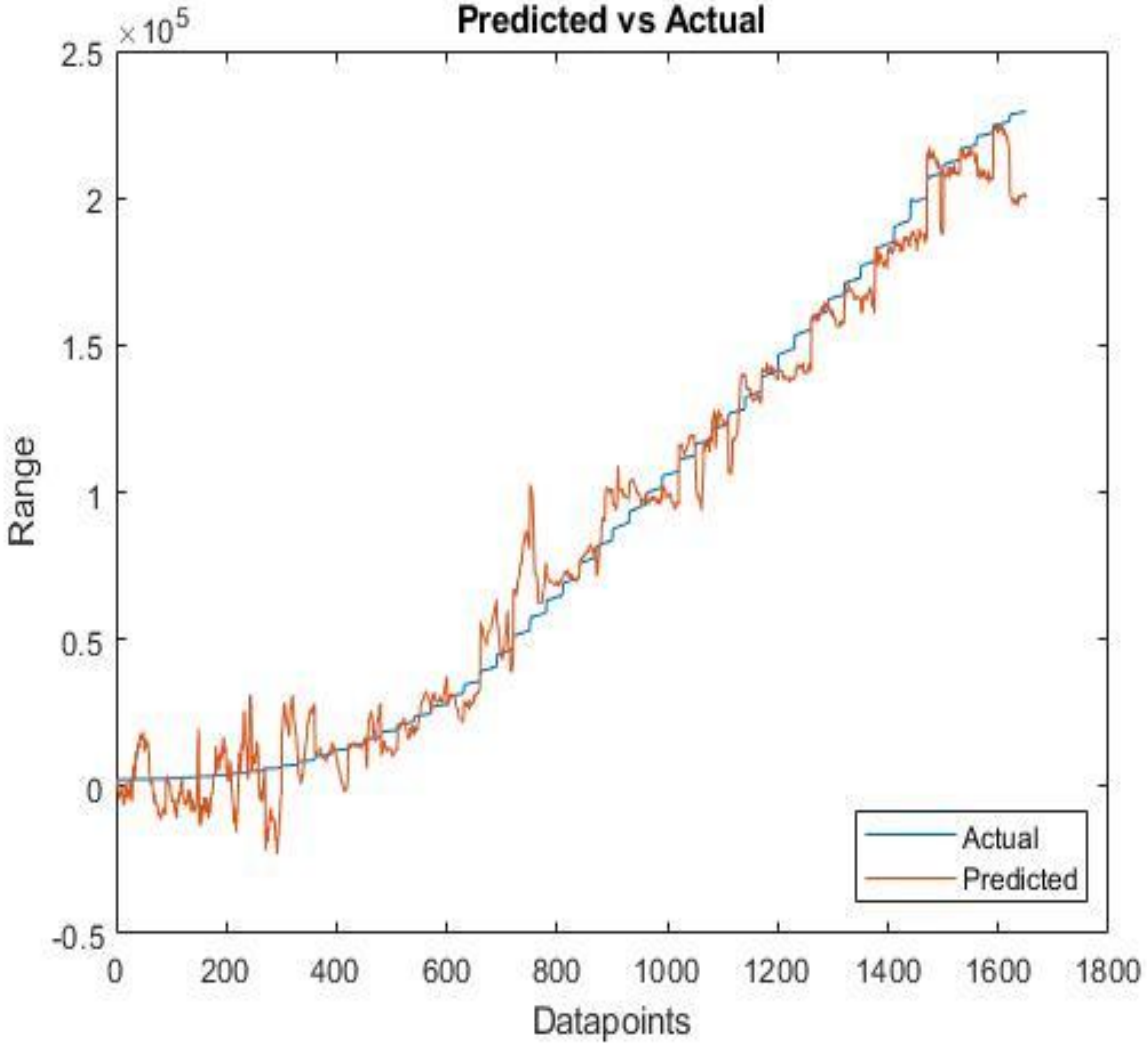


Inference -

We cannot make the depth of the architecture huge, as the model tends to overfit, as seen.

In this figure, after 2nd Epoch onward, the training RMSE decreases where as Validation RMSE increases.

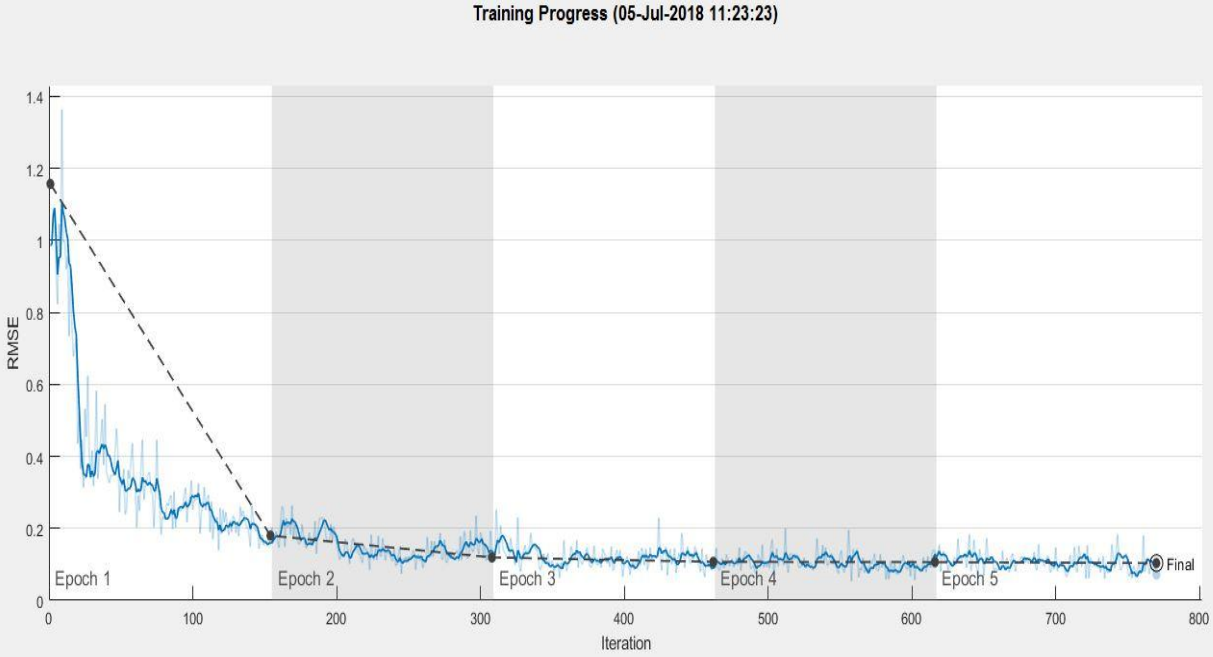
This suggests the network is memorizing the training cases, and is a bad solution.

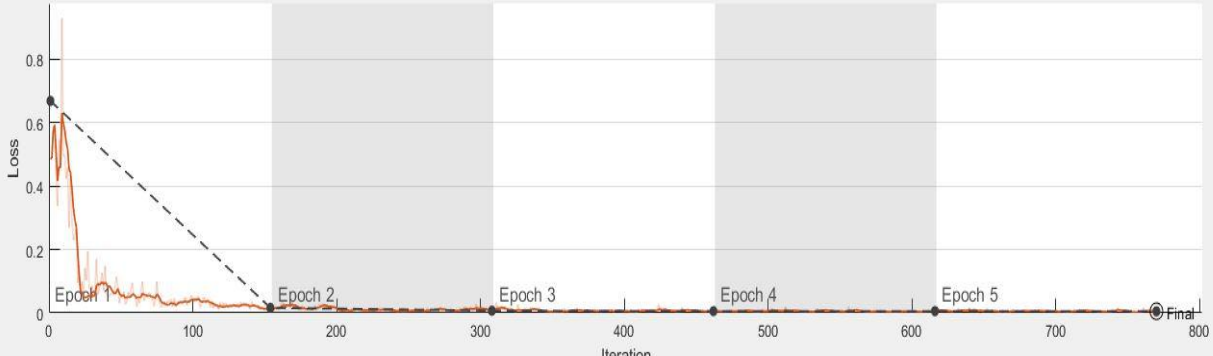


Inference -

After all the above observations, this is the best achieved results from the chosen architecture.

Key observation is that the CNN works very well for all the range values which was not the case earlier.





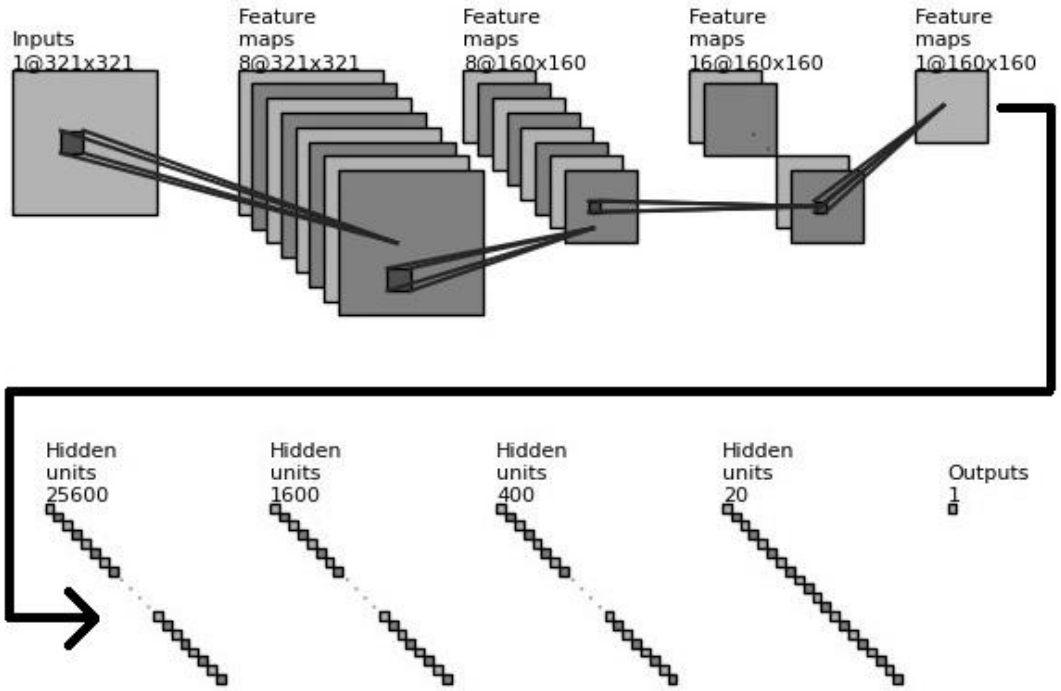
Inference -

Though we see a similar start in rmse and loss data as before, but the values goes to zero very fast and stays very low throughout the training.

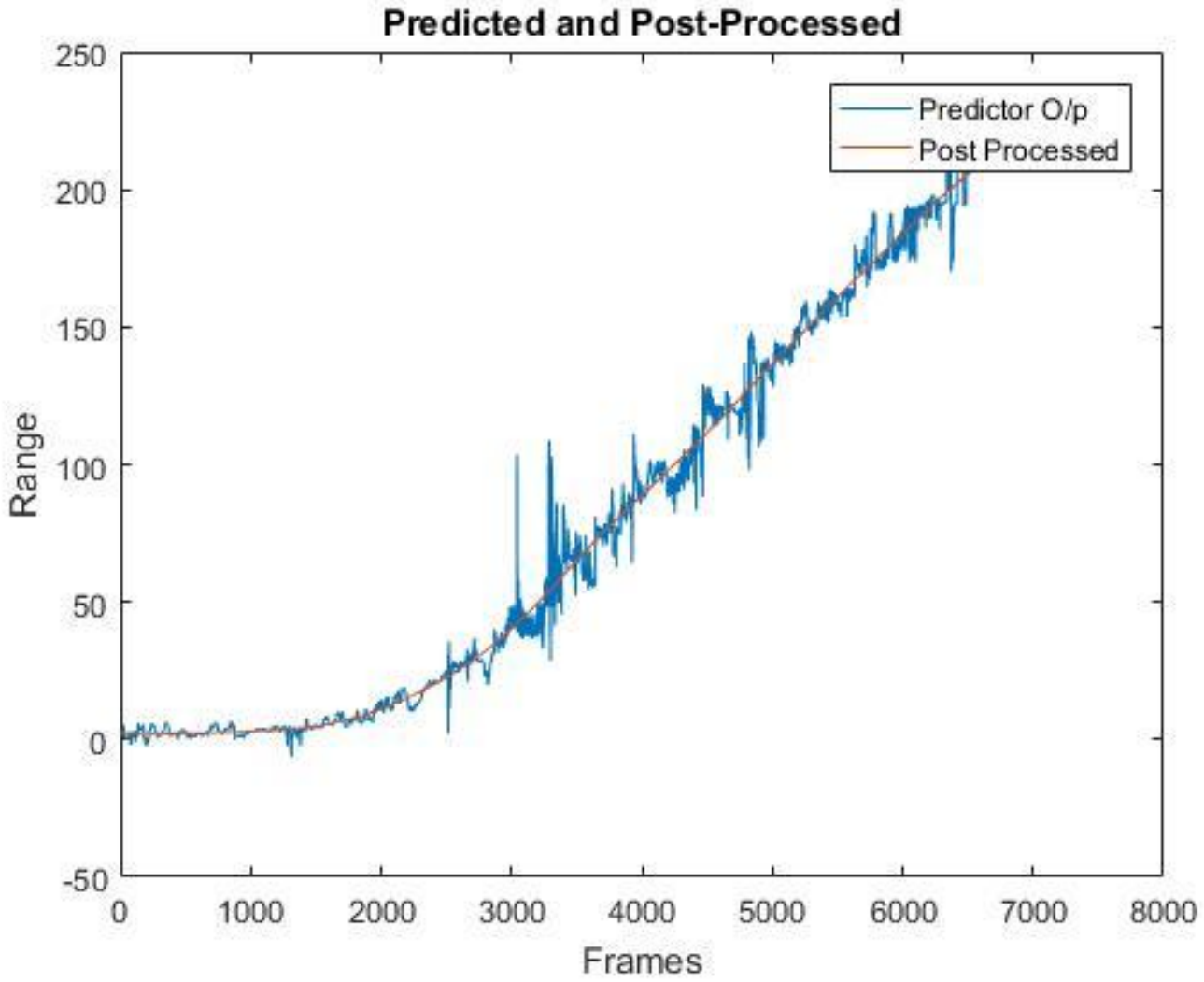
Key observation is that the CNN never goes into overfitting in this case.

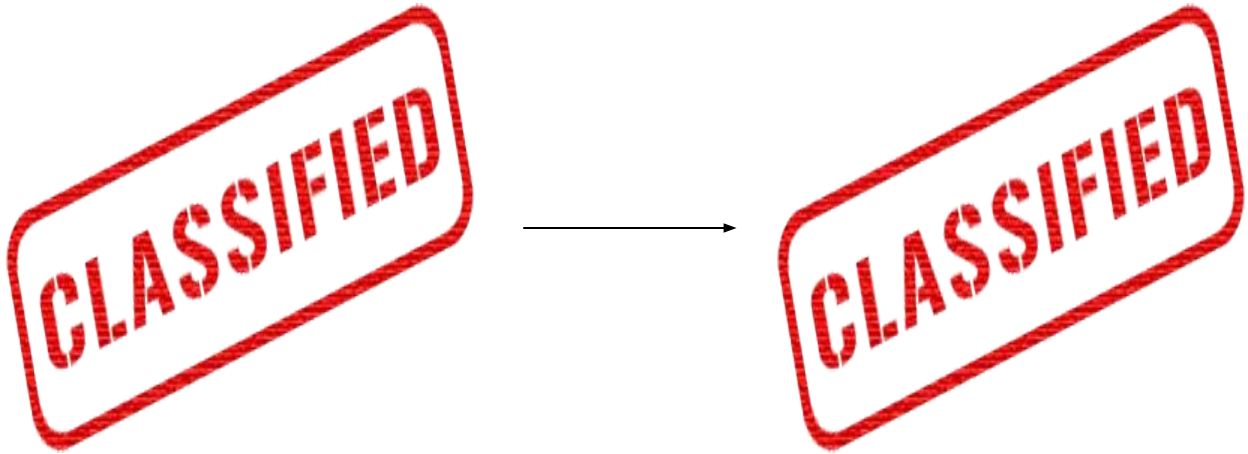
The RMSE value for this case was the lowest and equal to 10.35.

Final Architecture



We have come up with a script which takes a video input, and frame by frame regresses the range which is then stamped. The ranges which are stamped, are also plotted in the next figure!


Predicted and Post-Processed
Predictor O/p
Post Processed
Range
Frames
250
200
150
100
50
0
-50
0
1000
2000
3000
4000
5000
6000
7000
8000



Pre Convolution Post Convolution

The filters of the first layer have been visualised and as you can see in this image some kind of derivative has been taken. These are learned automatically by the CNN.

Where do we stand?

The CNN approach needs to be fine tuned.

The regressors and the data scrapers need to be combined for a working solution!

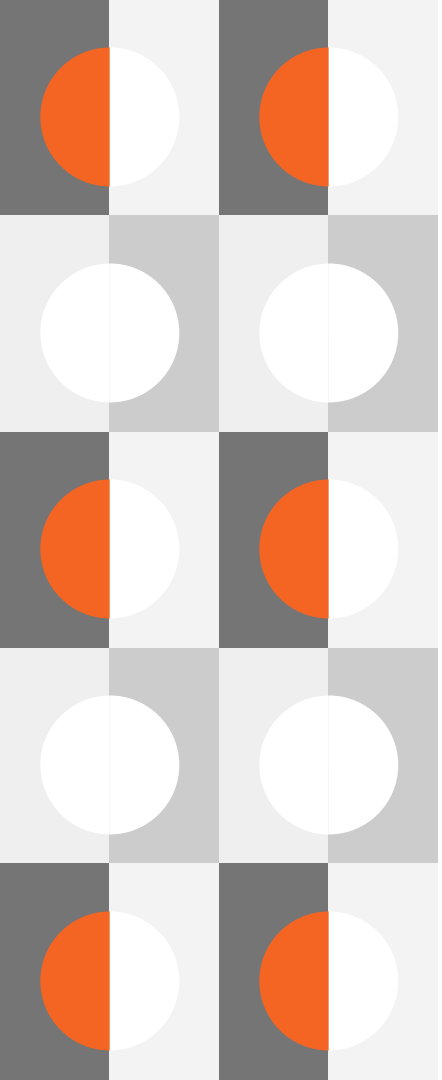
Advantages of CNN over traditional methods of range regression

- Fog
- Atmospheric aberrations
- Site selection
- Mode of Tracking
- IR fluctuations

Plan of Action

1. Fine tune the CNN, to operate for large ranges.
2. Use the Decision Tree Classifier for short ranges.
3. Find an optimal range separating the region of operation of both the methods.
4. Combine the results, and discuss merits and demerits

Timeline of Internship



21st May

Joining Date

Finding a department to intern in, and deciding the problem statement

1st June

Literature Survey

Started studying and finding about the existing system, explored the industrial usage

7th June

Data mining begins

Formation of dataset is essential for the part of ML/Deep Learning applications

15th June

Implementation of Methods

Various methods were implemented and inferenced parallely, the insight was important

5th July

Submission of Code

All codes, and data submitted back for evaluation and HDDs cleaned

The wheels are in motion...

We have also submitted a paper for publication in IEEE - Recent Advances in Intelligent Computational Systems (RAICS) - 2018.

Current Status - **Under Review**

Ms. Sasmita Mahakud, JRF at DRDO is pursuing this line of work and working on NN with memory to enhance our results, and we hope we can implement it sooner or later.


MINISTRY OF DEFENCE
DEFENCE RESEARCH & DEVELOPMENT ORGANISATION
बलस्य मूलं विज्ञानम्


विद्या शक्तिः प्रवर्धनम्
iist

Thank You