Study of Impinging Liquid in Scramjet Engine using Deep Learning Techniques

Prajith Kumar KP[#1], Bhartendu Thakur[#2], Ajith Kumar S [#1], Aravind Vaidyanathan. [#1]

[#1]*Department of Aerospace Engineering*
[#2]*Department of Mathematics*
Indian Institute of Space Science and Technology, Kerala, India.

prajithkumarkp1994@gmail.com, , aravind7@iist.ac.in

Abstract— The spray formation by impinging liquid jets introduced into supersonic cross-flow of Mach number 1.7 from a backward step is experimentally studied. Five injectors of 0.8 mm diameter inclined at angles 0, 15, 30, 45 and 60 degree each towards the central axis are used. Due to angled injection the fuel streams coming from both injectors will impinges and produces an elliptical liquid sheet perpendicular to the plane of impingement. The interaction of this sheet and supersonic airstream produce fine mist which was visualized using various laser optics techniques like Laser Shadowgraphy and Particle Master Shadowgraphy. Mie Scattering experiment was done to study the mixing characteristics. The results shows, high level of mixing and the formation of very fine droplets. Deep learning techniques were used to predict the best design conditions by adopting Variational Autoencoder method for generating new images with an extra Neural Net Block to generate corresponding design conditions.

***Keywords*— Scramjet, back step liquid injection, Neural Networks, Variational Autoencoders,**

I. Introduction

Human's desire to travel in supersonic speed can be made possible only through Scramjets. But the small residence time available for mixing and combustion of fuel and air in Scramjets act as major stumbling block in the technological development. Numerous studies were done on gaseous fuel though they cause problem in storage and availability there is limited information in liquid fuelled Scramjets. Limitation of carrying out experiment and the high time consuming Large Eddy (LES) simulations reduced the number of models that can be tested. This paper address these matters:

A. *By adopting a backward faced injector issuing doublet impinging liquid parallel to the supersonic flow*:

Due to angled injection the fuel streams coming from both injectors produce an elliptical liquid sheet perpendicular to the plane of impingement. The interaction of this sheet and supersonic air-stream produce sized particles which shows as mist in Micro-shadowgraphy.

A large number of studies were done on mixing enhancement of gaseous fuels like Hydrogen because of its superiority in combustion performance. However the lower density of hydrogen results large storage volumes. Thus there is a need to use hydrocarbon fuels, especially Kerosene due to the ease of storage and availability. However the liquid fuelled supersonic combustors are imparting many technical challenges compared to gaseous fuels. This include (1) a deeper fuel penetration into the air stream for better mixing; (2) generation of smaller liquid fuel droplets for faster evaporation; (3) appropriate flame stabilization mechanism for piloting and sustaining combustion; and (4) a substantial reduction in drag losses associated with processes of mixing and flame holding [1]. These challenges were removed using a backward step injector with parallel flow (Figure:1) because the parallel injection reduces the pressure loss and strong shocks compared to transverse injection but with reduced penetration height and droplet formation. Through the incorporation of doublet injection the above problems can be resolved.

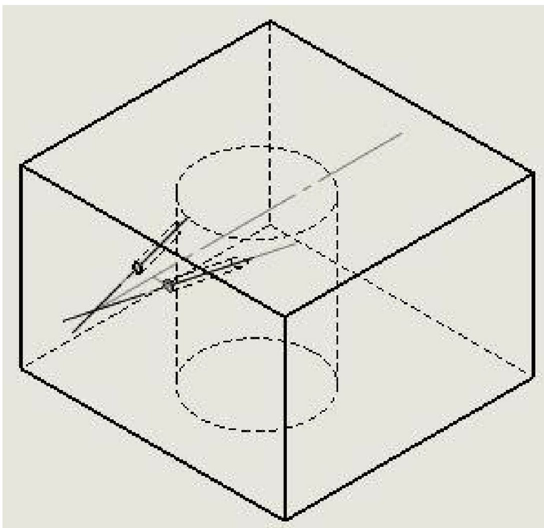


Figure 1: Injector Design

B. *Predicting an optimal design using Machine learning:*

Scramjets at Supersonic Speed suffers from inefficient mixing in Combustion-Chamber, but with optimal setup, mixing can be made efficient. To get optimal setup we proposed an analytical approach with prediction of optimal parameters using Neural Networks. In this proof of concept approach, we can use deep learning [2] to learn mappings from experimental images to various parameters like penetration height & spread, which have direct impact on improved mixing and flame stabilization to get design of a Scramjet Combustor having high efficiency.

The Deep Learning techniques applied to the problem defined above are Variational Autoencoders with a Generative Model, which broadly consists, Encoder, Decoder with convolution filters (both 1D & 2D), LeakyRelu activations and Dense Layers. Autoencoder network is actually a pair of two connected networks, an encoder and a decoder. An encoder network takes in an input, and converts it into a smaller, dense representation, which the decoder network can use to convert it back to the original input. Like in Convolutional Neural Networks (CNN), the convolutional layers of take in a large image and convert it to a much more compact, dense representation which later fed to fully connected layer to predict the outcomes. The encoder is similar, it is a network that takes in an input and produces a much smaller representation (the encoding), that contains enough information for the next part of the network to process it into the desired output format. Typically, the encoder is trained together with the other parts of the network, optimized via back-propagation, to produce encodings specifically useful for the task at hand. In Autoencoders, encoder generate encodings specifically useful for reconstructing its own input. The entire network is usually trained as a whole. The loss function is usually either the mean-squared error or cross-entropy between the output and the input, known as the reconstruction loss, which penalizes the network for creating outputs different from the input. As the encoding (which is simply the output of the hidden layer in the middle) has far less units than the input, the encoder must choose to discard information. The encoder learns to preserve as much of the relevant information as possible in the limited encoding, and intelligently discard irrelevant parts. The decoder learns to take the encoding and properly reconstruct it into a full image. Together, they form an Autoencoder.

Variational Autoencoders (VAEs), provides a probabilistic manner [5] for describing an observation in latent space and have one fundamentally unique property that separates them from Autoencoders, and that makes them so useful for generative modelling: their latent spaces are, by design, continuous, allowing easy random sampling and interpolation. It achieves this by doing something that seems rather surprising at first: making its encoder not output an encoding vector of size n, rather, outputting two vectors of size n: a vector of means, μ, and another vector of standard deviations, σ. Unlike typical Autoencoder, in VAE, encoder is formulated to describe a probability distribution for each latent attribute, which made it ideal for our approach.

With our method, if we can provide other experimental or computational results of the Scramjet studies done so far on any design it can be Pylon, Strut, Step or any shape with our approach, our model can be trained rigorously to predict an optimum design for given requirements. This may led to solve most of the designing problems associated with Scramjets. Once we have sufficient experimental data, we may report optimal design with our approach.

II. Experimental Set-up & results

The experiment was conducted on a supersonic wind tunnel of Mach number 1.7. The stagnation temperature is 300 K for all the test cases and the stagnation pressures were at 4 and 5 bar. The test section has an optical window of size 150x40 mm. Acetone was used for injection because its properties match with hydrocarbon fuels and its high volatile nature causes hardly any stains on the optical window. Experiments were conducted for all possible combination of the following Jet momentum ratio (J = momentum of jet/moment of air) and the impinging angles (θ).

J	0.0089	0.0139	0.0227	0.0351	0.0356	0.0551

(a) Ratio of momentum of jet & moment of air

θ	0°	15°	30°	45°	60°

(b)Impinging angles

TABLE 1: Test Conditions, (a) & (b)

The spray formation due to impinging jets in stagnant atmosphere was studied for deriving insights on the effects of supersonic flow on the atomization. The three (internal and external) forces which affect spray generation are hydrodynamic force before the jet collision, impact force due to the jet collision, and aerodynamic force. Figure: 3 shows the breakup pattern of like impinging jets at an angle of 45 degrees at mass flow rates and it can be categorized into three patterns a) Closed rim b) Open rim c) Fully developed spray at different velocities [3].

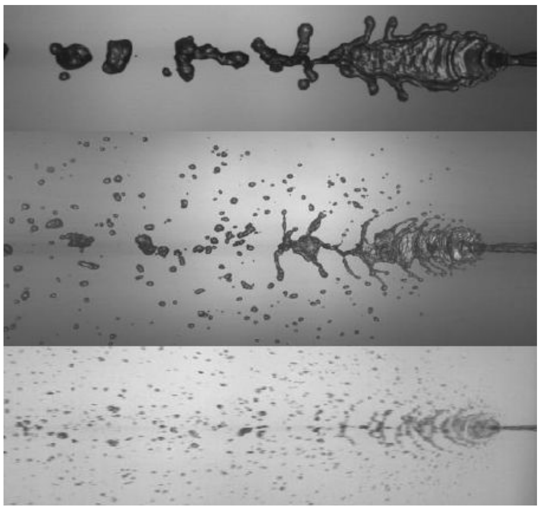
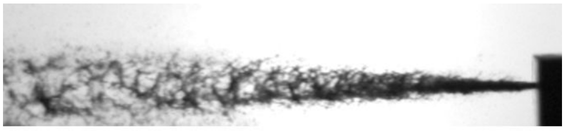
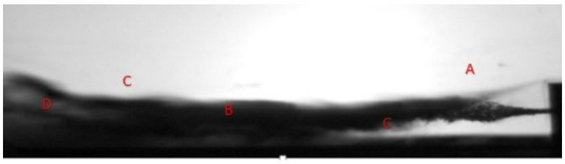


Figure 3: a) Closed rim b) Open rim c) Fully developed spray (Flow is from right to left)

All test conditions produces fully developed spray only. Due to mass conservation the liquid jets after impingement are squeezed into thick bulk liquids and in order to conserve momentum the thick liquid jet is stretched out into lateral direction. Even in the absence of supersonic-flows the liquid stream produced from the impingement point exhibits highly dispersive fluid motion. The impingement of the jets at higher velocities induces strong impact waves leading to the disintegration of liquid sheet. The interaction of liquid sheet and atmospheric air also amplifies the spray formation. Still from the image (Figure 3) it is clear that the breakup length is few millimetres. From figure (4.b) it is clear that impinging angle had a direct relationship with penetration height as well as mixing. Four regimes can be seen in the spray formed. They are a) Fine droplets in the rear shear layer, b) The dense spray core along the centre, c) Dilute outer and lower layer formed due to KH wave and by Prompt atomization[4] & d) Spray branching/ bifurcation. Apart from these the interaction of shocks and impingement also accelerated spray to form fine mist.



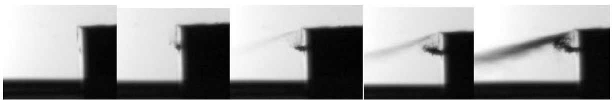
a



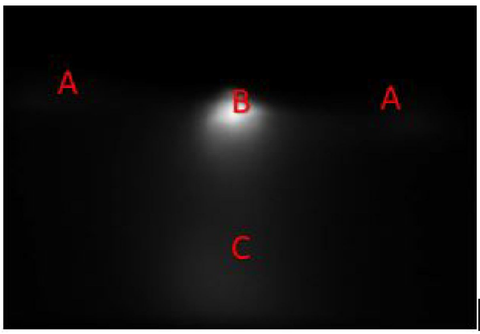
b

Figure 4: Spray Structure at 15 degree and J3 condition a) Without supersonic flow b) with supersonic flow

The time-sequence images (Figure: 5(a)) shows that the flow of drops to the shear layer is due to pressure difference. Mie Scattered images also shows this trend through two lobes (Figure: 5(b)). The spreading of drops or mixing was more when the angle of impingement increases.



(a) Images at a time interval of 0 to 800 μs



(b) Mie scattered image at x = 20mm

Figure 5: Experiments details

For analysing the result from Mie scattering a parameter called spread area (i.e. the number of pixels which shows intensity greater than 0.1 in normalised image) was introduced

Figure 6 shows that at location x = 5 mm from the injector 0 degree have less mixing. It is because 60°, 45° brings the flow close to the centre line (say converging effect) but due to impingement the flow become sheet (say sheet effect) like instead of circular and also the penetration height increases and also the backward flow of the liquid to the shear layer and the recirculation (say geometry effect) all cause improved spread area in 60° and 45° injectors. Geometry effect is common for all injectors. Now comparing 45° and 60° the former one shows high mixing because 60° have more converging effect compared to 45°.

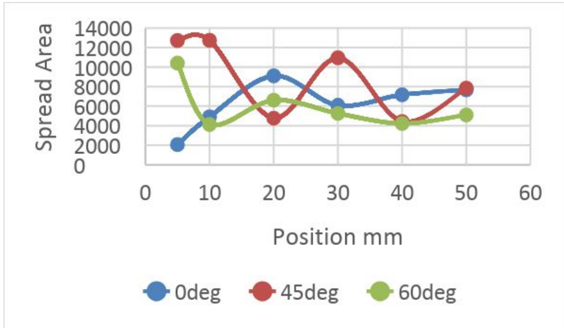


Figure 6. Spread area versus Position for J=0.0139 (main flow pressure 3 bar and liquid injection Pressure 2 bar)

After 5 mm the mainstream air momentum is high enough compared to the jet momentum due to this the penetration of liquid becomes difficult hence the convergence effect overtakes the sheet effect leading to decreased mixing for higher degree injectors. This effect will be vice versa near to the injector because of step the momentum of air near to it will be less.

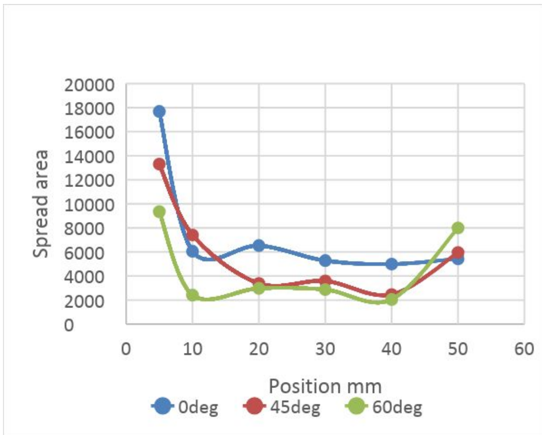


Figure 7. Spread area versus Position for J= 0.0089 (main flow pressure 3 bar and liquid injection pressure 2 bar)

Figure 7, shows the result of Mie scattering at J =0.0089.Here spread area is more for 0, 45 then 60 at x=5mm it is due to the high momentum of air at pressure 5 bar absolute causes the coalescence of the liquid and reduces the sheet effect together with convergence effect contribute to the above trend in spread area for high pressure flow. This trend is continued for other positions also.

III. Application of deep learning

To study the applicability of Deep Learning techniques in defined problem, we trained convolutional neural network (NN_1) to classify J, θ pairs for each experimental images. The trained CNN performed better than expected and reported accuracy for more than 99% with Categorical Cross entropy loss $7.0694e^{-06}$ for training and $2.9159e^{-05}$ for validation. This performance proved the applicability of Deep learning for further applications.

Our Method is described in two parts, first part is application of Variational Auto-Encoders to encode images in 3-Dimensional Latent space which is further fed to a Neural Network (NN_2) to learn J, θ values as shown in Figure 8.

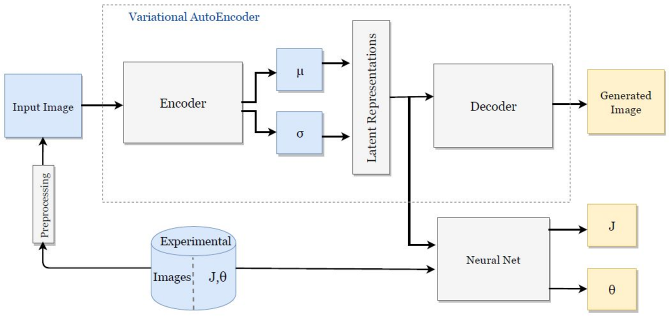


Figure 8. Block Diagram: VAE concatenated with NN_2

The model essentially should produce 3-Dimensional Latent values which can reproduce images as well as J, θ values. To reproduce the images we feed Latent values as input to Decoder & to reproduce J, θ values for those images, we feed Latent values as input to Neural Network (NN_2). This basically means that with having different Latent values our model can generate the near to actual experimental images with corresponding J, θ. This led to second part of our model. In this part we use architecture as in figure 10, to generate different images & and its corresponding J, θ with different Latent values.

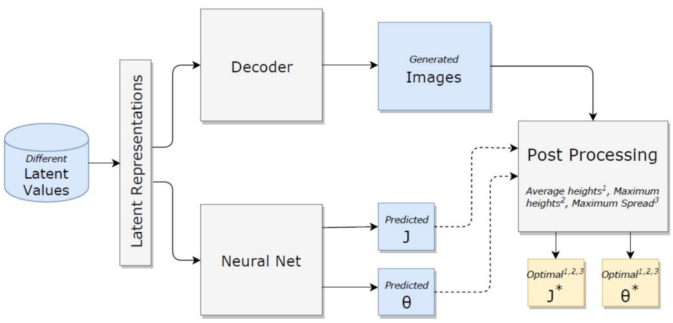


Figure 9. Block Diagram: Generative Model

A. *Data Generations & Processing*

1) Pre-processing

We have a total of 25626 instances of images and their parameters {J, θ}. We've randomly selected 80% of the instances for training and remaining 20% for validation / testing and hyper parameters tuning. Raw images are captured using high-speed camera, which has small shift and scale errors. These errors has been removed by selective cropping as shown in Figure 10.

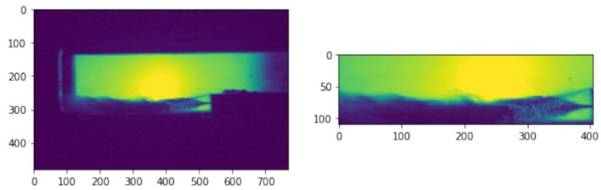


Figure 10. Image Pre-processing

2) *Data Generation*

With generative model, we generated a total of 27000 instances of images and their parameters {J, θ} by varying every dimension of Latent representations. For every dimension (with mean μ_i & standard deviation σ_i) of Latent representation we sampled values from U_i, where U_i is a uniform distribution over [μ_i -2 σ_i , μ_i +2 σ_i] respectively. Some generated images are shown below:

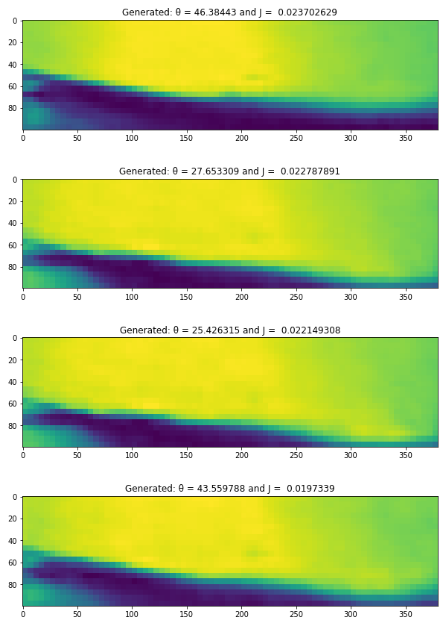


Figure 11. Images from Generative Model

3) Post-processing

Boundaries of generated images are retrieved and retained using Image processing techniques. These Image processing techniques includes Image Binarization followed by tracing the exterior boundaries using Moore-Neighbor tracing algorithm modified by Jacob's stopping criteria. Some images after boundary retrieval are shown in figure 12.

These boundaries were later used to calculate (a) Average height, (b) Maximum height, (c) Area of spread per image for reporting the optimal J*, θ*.

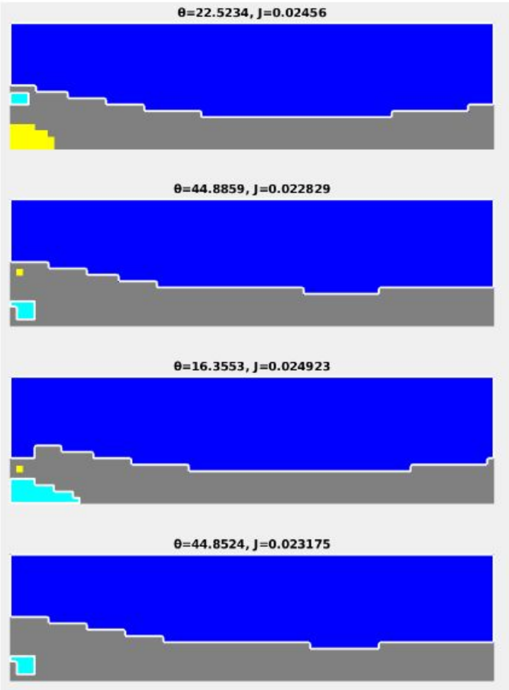


Figure 12. Generated images boundaries

3) Network Architectures

A total of 4 Network Architecture were developed (a) Encoder, (b) Decoder, (c) Variational Autoencoder (d) Neural Net (NN_2). **Variational Autoencoder** discussed in previous section is basically concatenation of Encoder, to variational layer to Decoder (as discussed below):

(a) **Encoder** are built using seven Convolutional (2D) layers, three Max-pooling layers and two Dense layer with most of the activations are either Relu or LeakyRelu[6].

(b) **Decoder** is in generally build in the same fashion as of encoder. In Decoder we have used seven Convolutional (Transpose) layers, three Upsampling layers and two Dense Layer with most of the activations are either Relu or LeakyRelu.

The architectures for Encoder & Decoder is inspired from Squeeze-Net[7] and Inception-Net . As depth of these architecture is high, it is not discussed in this paper [1].

(c) **Neural Net (NN_2):** As Discussed before, we are using a neural net to learn mapping between, 3D-Encoded Latent values to corresponding J & θ values. NN_2 a small network of 1D-Convolutional and dense layers as shown in figure:

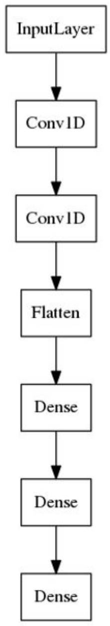


Figure 13. NN_2 Architecture

4) Optimization

We train our both networks parameters to minimize the mean squared error between the predicted outcomes and the ground truth values and optimize the weights with an adaptive learning rate calculated using the Adam[8] optimization function. Adaptive Moment Estimation (**Adam**) works well in practice and compares favourably to other adaptive learning method algorithms as it converges very fast and the learning speed of the Model is quiet Fast and efficient and also it rectifies every problem that is faced in other optimization techniques such as vanishing Learning rate, slow convergence or High variance in the parameter updates which leads to fluctuating Loss function. Also to avoid overfitting we have used concept of early stopping. In Early stopping, if after number of epochs with minimum change in the quantity (say loss) to be monitored (to qualify as an improvement) has not been observed then training is stopped.

5) Training Results

In this section we describe our results in training the networks and examine the features the networks learned. It is important to note training takes an average of 3ms/step on an Intel Xeon(R) CPU E5-2620 v4 @ 2.10GHz (x16), with Nvidia Quadro P2000 GPU, running the Keras framework with Tensorflow backend. Our fully trained network generated 27000 images with dimension (100, 380) in less than 1 min 30 seconds.

[1] Encoder, Decoder, VAE architecture are available at https://github.com/matrixBT/Scramjet-Learning

We trained VAE model with learning rate of 0.00025, for more than 50 epochs which took 84 secs per epoch, training history is shown in Figure 14.

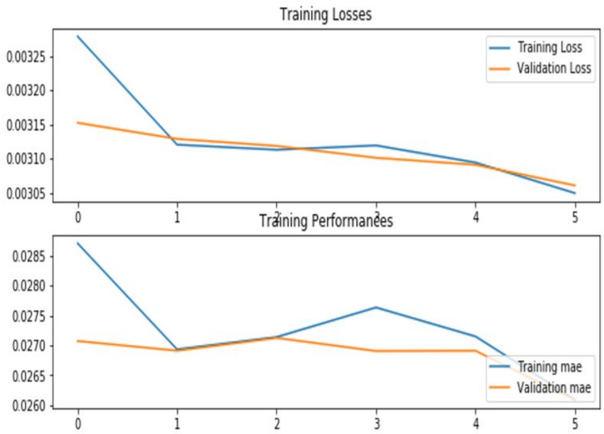


Figure 14. VAE: Training Losses in Last 5 epochs

Input image (pre-processed experimental) are encoded in three dimensional Latent space is shown in Figure 15, it’s notable that images with similar J, θ are in the same likelihood.

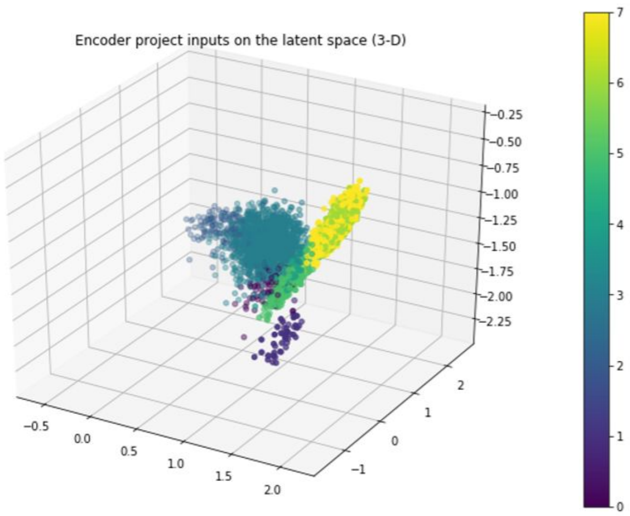


Figure 15: Latent Dimensional visualization
(Encoded outputs)

This proves that our model has learnt well, and we can expect decoder performance to be high as well. Thus we generated the images using decoder and compared it to the corresponding original images, Comparisons are shown in Figure 16

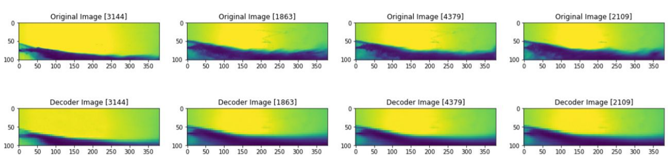


Figure 16. Output of Decoder

We trained Neural Net (NN_2) for 75 epochs to learn mapping from Latent representation to J, θ, with learning rate of 0.001, which took less than 1 sec per epoch, training history is shown in Figure 17.

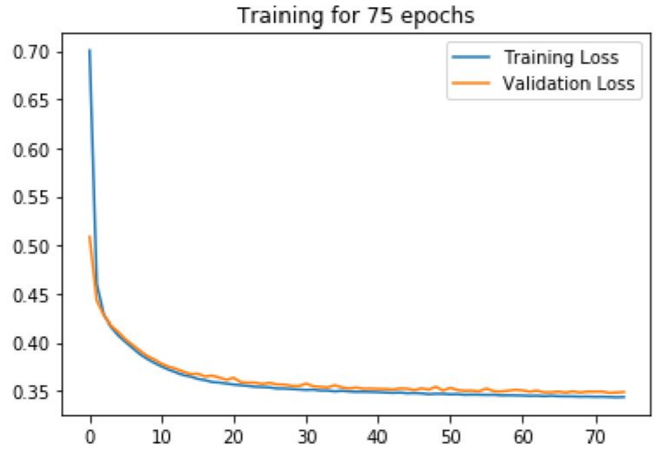


Figure 17: NN_2: Training Loss in 75 epochs

The training losses and the results above shows that, our model has performed well, and therefore images & J, θ generated for some different/new Latent values through this model can be considered close to the experimental images & J, θ.

IV. Results & Conclusion

An impinging jet from a backward step to supersonic flow was studied using various experiments. Through our machine learning approach we generated 27000 test conditions and used it to optimise the design parameters like J & θ. Here maximum penetration height, maximum average height and spread area are considered as performance parameters. It can be any parameter like Mixing efficiency, Maximum pressure loss, Combustion efficiency or Thrust and Specific Impulse because all these features are hidden in the images generated.

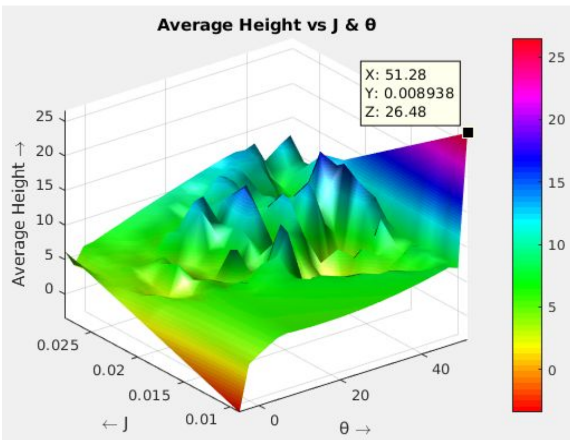


Figure 18. Surface plot of average penetration height at generated J & θ

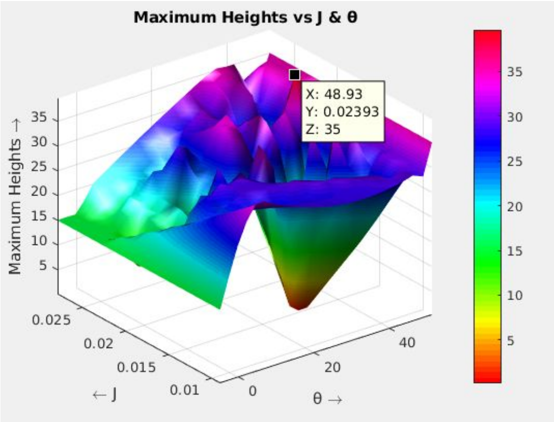


Figure 19. Surface plot of maximum penetration height at generated J & θ

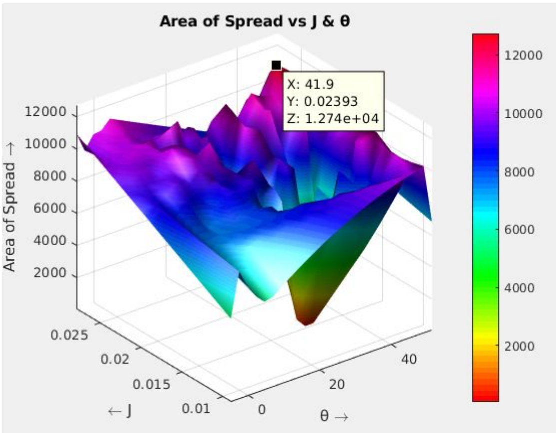


Figure 29. Surface plot of area of spread at generated J & θ

According to the average height the optimal design is having 51.28 degree as θ and 0.008938 as J and for maximum penetration height the conditions are 48.93 degree and 0.02393 J while considering area of spread 41.9 degree and 0.02393 J. The trend in the plots shows that when impinging angle increase the optimising parameters are increasing this is vivid from the experimental results too.

References

[1] Liyin Wu, Zhen-guo Wang, Qinglian Li, and Jiaqi Zhang"Investigations on the droplet distributions in the atomization of kerosene jets in supersonic crossflows," APPLIED PHYSICS LETTERS 107, 104103 (2015)

[2] Monfort, Mathew, Luciani, Timothy, Komperda, Jonathan, Ziebart, Brian, Mashayek, Farzad, and Marai, G Elisabeta. A deep learning approach to identifying shock locations in turbulent combustion tensor fields. In Modeling, Analysis, and Visualization of Anisotropy, pp. 375–392. Springer, 2017.

[3] S. S. Lee1, W. H. Kim and W. S. Yoon, " Spray formation by like-doublet impinging jets in low speed crossflows

[4] Tan, Zu Puayen, Bibik, Oleksandr, Shcherbik, Dmitriy, Zinn, Ben T, and Patel, Nayan. The regimes of twin-fluid jet-in-crossflow at atmospheric and jet-engine operating conditions. Physics of Fluids, 30(2):025101, 2018.

[5] Auto-Encoding Variational Bayes, Diederik P Kingma, Max Welling, 2013

[6] Empirical Evaluation of Rectified Activations in Convolutional Network Bing Xu, Naiyan Wang, Tianqi Chen, Mu Li, 2015

[7] SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer, 2016

[8] Adam: A Method for Stochastic Optimization, 2015, Diederik P. Kingma, Jimmy Ba