**Artificial Intelligence Enforced Impact Location Prediction**

**Aparna Mathew Joseph1 (416404), Tanvi2 (Matr..), Sajjala Sreekar Reddy3****(415917), Shiva Madha4(415932), Atharva5(Matr..)**

**1, 2, 3, 4 MSc. Computer-Aided Mechanical Engineering (CAME)**

**5 MSc. Management and Engineering in CAME**

Localization of an impact on an aircraft nose cone is vital to assess the risk of damage. Collecting sensor data from the impact zone and passing it through an Artificial Intelligence (AI) have a better chance of predicting the impact location. This study trains various AI models on computer simulation data to predict when given experimental data. The team presented 5 different AI models out of which modified Alex Net (mAlexNet) architecture has the best predictions. This report summarizes the extension of project B; post-processing to improve prediction and compare different CNN architecture predictions of an experimental dataset. With the help of post-processing, the prediction from the mAlexNet is narrowed down to the exact impact location. Two new architectures have been adapted to better fit the dataset: Residual Network and VGG.

**KEYWORDS:** impact, regression, post-processing, CNN, ResNet, VGG

**INTRODUCTION:** The next generation of aircraft must comply with low carbon emission regulations. As structural weight is a key component in fuel consumption, it is natural to change from metallic to lighter composite structures. However, their conservative damage tolerance design needs to be revised to fully utilise their strength to weight ratio. One way of ensuring the safety of the structure without overdesigning the components is to have continuous monitoring of its state to detect any alarming external impact events which could lead to loss of strength at the early stages of evolution (Tabian I, et al., 2019). With this motivation, a project regarding the prediction of impact location was undertaken. Previously, CNN along with other numerous models was implemented. Compared to other models, the main advantage of CNN is its ability to automatically detect patterns without any human supervision, which made it the most used ([Laith Alzubaidi](https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8#auth-Laith-Alzubaidi), et al., 2021). It was found that the CNN model gave the closest predictions. One of the most notable architectures, namely AlexNet was implemented. AlexNet consequently improves the CNN learning ability by increasing its depth and implementing several parameter optimization strategies ([Laith Alzubaidi](https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8#auth-Laith-Alzubaidi), et al., 2021). Through its hyperparameters’ tuning, a mean square error of ~16.3 was obtained. Further for the improvement of the proposed models, certain prospects were considered. Firstly, running post-processing to improve the prediction. Secondly, to compare different CNN architectures (tuned to fit the dataset using Keras Tuner).

**METHODS:** An AI model to detect the impact location on a 150mm x 150mm impact surface (domain) was performed on a dataset obtained from a computer simulation and validated against experimental data. The initial training dataset contains sensor data at 244 locations which are 5mm apart in the x- and y-axis of the top right quadrant of the domain (75mm x 75mm). Data augmentation was performed to expand the initial dataset to cover the entire domain. Data pre-processing followed next for both training and experimental data to cut out the noise of ~20,000 and ~100,000 data points respectively in the start and additional sensor information post the first impact wave. Later, the training and experimental data were resampled from ~18,00 and ~300 data points (respectively) to 100 whilst maintaining the shape of the sensor signal to reduce the input shape of the AI model.

Further, the training dataset was split into a training and testing set. Various models were created for the first part of project B, such as Feed Forward Neural Network (FFNN), modified AlexNet (mAlexNet), Support Vector Regression (SVR), and K-Nearest Neighbours. Tuning of an AI model is crucial to fit the architecture to the dataset and for this project, several hyperparameters were chosen to be tuned for the best fit. For convolution layers, dense layers, and model compilation: (layers, filters, kernel size, strides, activation function, pooling layers, pooling size, regularization count and size, and padding), (dense layer count, neurons, and activation functions), and (loss function, and optimizer) were chosen respectively. Hyperparameters tuning was performed for FFNN and mAlexNet models using Keras Tuner. It was then tested and validated on the respective datasets. Out of all the models, mAlexNet produced the best accuracy for training and experimental data.

Post-processing: Results from the implemented mAlexNet model are promising with an average error (~3.6mm). So, employing a post-processing step seems rationale for the given dataset. Considering the 925 training coordinates, an algorithm was built to locate the closest training data point to the predicted coordinate and create a buffer set of 25 points encompassing 2 layers around the closest training point as shown in Figure-2(a).

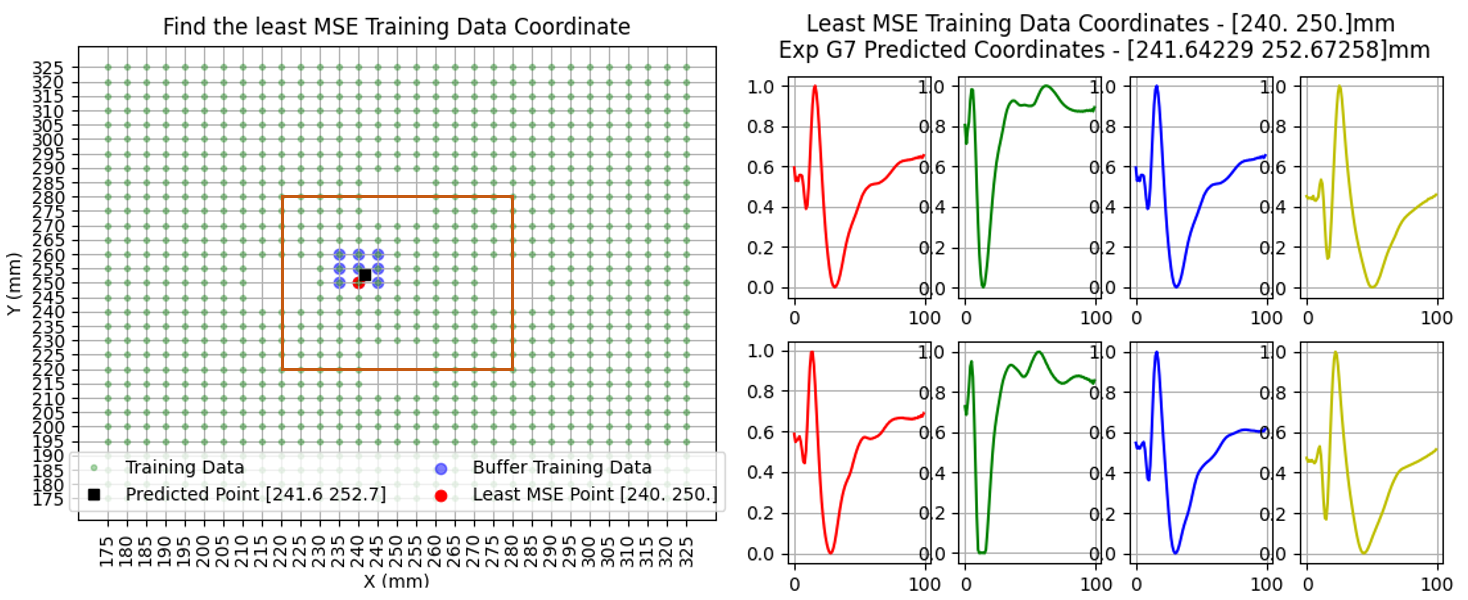


Figure-2 (a) - Buffer set surrounding predicted point located in the inner region.

(b) - Sensor data comparison between the predicted and least MSE point.

Considering some missing data at the sensor locations, the size of the buffer set is varied between 1 and 9 coordinates (1 being the closest training coordinate itself). The MSE of the training data in each location of the buffer set and the experimental sensor data is taken and the training coordinate that yields the least is determined as the real impact location.

To evaluate the efficiency, the output from the post-processing step is compared with the ground truth. 15 out of 18 coordinates were pointed correctly with 3 incorrectly pointing the y-axis coordinate by 5mm.

Experimental data has 3 different sets of coordinates: increased impact velocity, inner, and outer region coordinates. The implemented mAlexNet model does not include the aforementioned features. The team strongly believes that training the model with more features would yield a better outcome after post-processing.

Investigation of different architectures: Initially deployed mAlexNet had the kernel size modified from 11 to 5, to make the architecture more compatible with the given dataset, which proved to be effective with an increase in accuracy. This made us pursue the multi-layer models of the smaller kernels and filter sizes. It was experimentally proven that the small-size filters made the receptive field similarly efficient to the large-size filters in a parallel assignment. The modified VGG implemented thus has an input layer with kernel size 3 with convolution layers of kernel size 1 to regulate the complexity of the network. VGG also addresses another very important aspect of CNNs: depth. VGG stands for Visual Geometry Group; it is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers. The "deep" refers to the number of layers with VGG-16 or VGG-19 consisting of 16 and 19 convolutional layers. Fig1 shows the VGG network, which was implemented for our dataset.

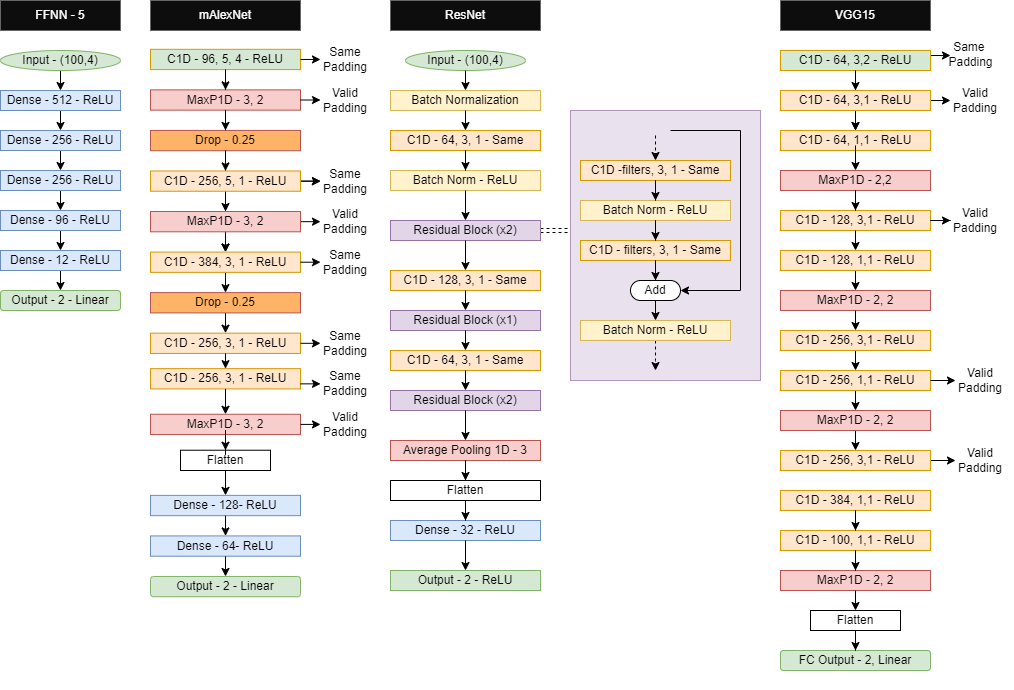


Fig1: Network architecture implemented (FFNN, DCNN, ResNet and VGG)

Studies have shown that the accuracy decreases by adding more layers to the network, causing the vanishing gradient problem. As we make the CNN deeper, the derivative when back-propagating to the initial layers become almost insignificant in value(1 Aqeel Anwar, 2019). Residual Network (ResNet) addresses this problem, the network can be shallower with the same accuracy or improved accuracy. Shallower network means: Number of layers can be reduced, training time can be shorter as well and a better dropout is also investigated. Fig1 shows the implemented Wide ResNet for the given training dataset.

In VGG, as the number of layers increases in CNN, the ability of the model to fit more complex functions also increases. Hence, more layers promise better performance. However, since the weights of the model are updated through the backpropagation algorithm, minor changes to each weight may lead to a decrease in the loss of the model. The model updates each weight so that it takes a step in the direction along which the loss decreases. This is nothing but the gradient of this weight which can be found using the chain rule. However, as the gradient keeps flowing backwards to the initial layers, the value keeps increasing by each local gradient. This results in the gradient becoming smaller and smaller, thereby making changes to the initial layers very small. This, in turn, increases the training time significantly. The problem can be solved if the local gradient becomes 1. This can be solved by ResNet since it achieves this through the identity function. So, as the gradient is back-propagated, it does not decrease in value because the local gradient is 1(3).

**Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Depth(layers) | MSE (Mean Squared Error) (mm) | Validation Accuracy | Run-time (seconds) |
| SVR |  | 23.7 |  |  |
| KNN |  | 23.5 |  |  |
| FFNN | 6 | 39.43 | 0.9568 | 18 |
| mAlexNet | 8 | 16.3 | 0.9838 | 46 |
| VGG | 15 |  | 0.9820 | 100 |
| ResNet |  | 60 | 0.8925 | 70 |
| Wide ResNet | 10 | 13 | 0.9688 |  |

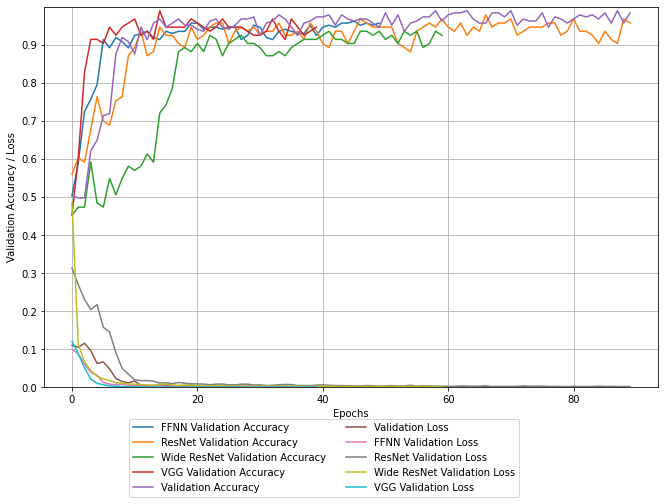


Fig3. Graph for Validation Accuracy/ Loss versus Epochs for FFNN, ResNet, Wide ResNet, VGG.

**Discussion**

- How does FFNN have better testing accuracy but doesn’t work as much for experimental

- AI, ML models prediction of experimental data

- Does post-processing predict better

-

**CONCLUSION:**

Which neural network is the best and why?

Possible extensions to this work (include impact velocity, region separation)

**Discussion**

From Fig3, we come to know that the

References

1. [Anwar](https://aqeel-anwar.medium.com/?source=user_profile----------------------------------------), Aqeel.”Difference between AlexNet, VGGNet, ResNet, and Inception”.[Towards Data Scienc](https://towardsdatascience.com/?source=post_page-----7baaaecccc96-----------------------------------)e,7th June 2019.
2. Tsang, Sik-Ho.”Review: WRNs — Wide Residual Networks (Image Classification)”.[Towards Data Scienc](https://towardsdatascience.com/?source=post_page-----7baaaecccc96-----------------------------------)e,1st December 2018.
3. <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/>

Iuliana Tabian, Hailing Fu, Zahra Sharif Khodaei (2019). A Convolutional Neural Network for Impact Detection and Characterization of Complex Composite Structures. Sensors 2019, 19(22), 4933.

Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie , Laith Farhan (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data volume 8, Article number: 53.

1. <https://theaisummer.com/cnn-architectures/>

For improving the implemented CNN model, its hyperparameters need to be tuned till the performance improves. The following are few of the hyper-parameters that were used under Conv1d:

N hidden layers

N filters

Kernel size is a number specifying both the height and width in the (square) convolution window (G. Kalyani et al, in Trends in deep learning methodologies 2021). It controls the speed of transition and can be optimally determined to best fit the

data through the kernel density estimation connection (Adaptive Learning Methods for Nonlinear System Modeling, 2018).

Strides denotes how many steps we are moving in each steps in convolution ([dshahid380](https://medium.com/@dshahid380?source=post_page-----cb0883dd6529-----------------------------------), Towards Data Science, 2019)

Activation function used was ReLu, as it was found to speed up the convergence of neural networks when compared to sigmoid or tanh functions. This advantage is because of its non-saturating form. ReLU can be implemented by thresholding an input of activation at zero. The ReLu is given by the equation below: f(x) = max (0, x) (Pratama Kevin, et al, International Journal of Advanced Smart Convergence Vol.6 No.4 73-79, 2017).

Dropouts is a technique that is utilised to reduce a model’s potential to overfit. It works by adding a probability factor to the activation of neurons within the layers of a CNN. This probability factor indicates the neuron's chances of being activated during a current feed-forward step and when it is involved in the process of backpropagation([Richmond Alake](https://richmondalake.medium.com/?source=post_page-----46c7974b46fc-----------------------------------), Towards Data Science, 2020).

Dropout size

Padding type

Max pooling is a pooling operation that calculates the maximum, or largest, value in each patch of each feature map([Jason Brownlee](https://machinelearningmastery.com/author/jasonb/) ,[Deep Learning for Computer Vision](https://machinelearningmastery.com/category/deep-learning-for-computer-vision/), 2019)**.**

**ResNet**

For 1 epoch:

1st Batch Normalization

2nd Conv1d with relu BN

1. Image
2. why resNet ?
3. how is it significant for time series net
4. how is different from trad CNN (Method)

**TITLE**

**Figure SEQ Figure \\* ARABIC 1 This is an example figure caption.**

**Sreekar1 (Matriculation Number), Tanvi Chinnapa(415947)2 (Matriculation Number),..**

**1Study Program 1**

**2Study Program 2**

Use this template to create your final report. It supposed to elaborate a new research question on the data you used in one of the two projects. The report includes the sections given below and an abstract at this place. There is one example for a report given. The report must not exceed six pages including references. Do not use more than five figures/tables. Stick to the template.

**KEYWORDS:** 10 pt, up to 5 keywords separated by comma.

**INTRODUCTION:** Here comes your introduction, no subheadings

**METHODS:** Here come the methods used

post-processing - (training results)

VGG14 - (why VGG, modifying the architecture to fit the dataset

ResNet - (why ResNet, modifying the architecture to fit the dataset

**RESULTS:** Fill in your results, run time, validation results

post- processing

AlexNet, VGG, ResNet, SVR, KNN (which is best performing model)

Accuracy,

Experimental results (comparison bar graph)

**DISCUSSION:**

Comparison between AI ML models

**CONCLUSION:**

Which neural network is the best and why?

Possible extensions to this work (include impact velocity, region separation)

**REFERENCES**

1. **CNN1 (Alexnet)**
2. **CNN2**
3. **Post processing**
4. **Comparison between CNN1 and 2**
5. **Different architecture comparisons**

**Methods & conclusion** :**Tensor board hyperparameter table.**

Previously , (one para on the method)

Decisions on hyperparameters.

the deviation of the predicted coordinates from the exact coordinates was plotted. The

I plotted the way the prediction is off from the exact coordinates

plotted the predicted point and its surrounding points. Calculated the MSE at all the surrounding points and defined the least MSE point to be the impact location

Based on this result we extended

1.Compare different CNN architectures  
(HP tuned). Motivation is to understand  
what kind of traditional architecture  
works best and how it could be modified  
to yield better accuracy using hyper-  
parameter tuning.  
2. run post-processing to better the  
prediction. we select a few surrounding

training data from the predicted location

and compare the mse of sensor data

between training and experimental data,

choose the least mse position to be the

impact location.