COMPARATIVE STUDY OF ARTIFICIAL INTELLIGENCE MODELS IN IMPACT LOCATION PREDICTION

Aparna Mathew Joseph[1] (416404), Tanvi Chinnapa[2] (415947), Sajjala Sreekar Reddy[3] (415917), Shiva Madha[4] (415932), Atharva Betawadkar [5] (404949)

[1, 2, 3, 4] MSc. Computer-Aided Mechanical Engineering (CAME)
[5] MSc. Management and Engineering in CAME

Localization of an impact on an aircraft nosecone is vital to assess the risk of damage. Collecting sensor data from the impact zone and passing it through an Artificial Intelligence (AI) has a better chance of predicting the impact location. This study trains various AI models on computer simulation data to predict experimental data. Various AI models were implemented out of which modified Alex Net (mAlexNet) architecture showed relatively better predictions. This report summarises the extension of project B; aiming to improve the impact location prediction, utilising post-processing, and investigating different Convolutional neural network (CNN) architectures. 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 (ResNet) and Visual Geometry Group (VGG).

KEYWORDS: Impact prediction, post-processing, ResNet, VGG, Alex Net

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 redesigning 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 (Luliana Tabian, 2019). With this motivation, a project regarding the prediction of impact location was undertaken. Previously in project B, CNN along with other numerous models were implemented. Compared to other models, the main advantage of CNN is its ability to automatically detect patterns without any human supervision, which makes it the most used. It was found that the CNN model gave the closest predictions. One of the most common architectures, namely AlexNet, was implemented. AlexNet consequently improves the CNN learning ability by increasing its depth and implementing several parameter optimization strategies (Alzubaidi Laith, 2021). Through its hyperparameter tuning, a mean square error of 16.3mm was obtained. Further, with the improvement of the proposed models, certain prospects were noted. Firstly, it was decided to compare different CNN architectures and their hyperparameters' tuning. Secondly, running post-processing to improve the prediction.

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, regularisation 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.

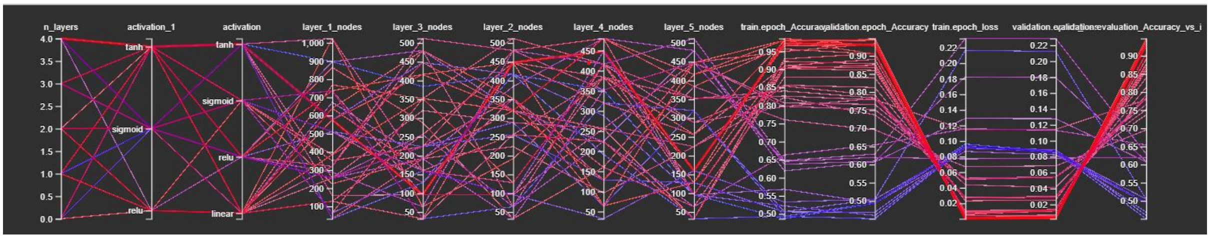

Figure 1. Hyperparameter tuning using Keras Tuner (visualized in Tensorboard)

To further improve the impact locations' prediction, post-processing was carried out for the mAlexNet architecture. Based on the 925 training coordinates, an algorithm was built to locate the closest training data point to the predicted coordinate and a buffer set of 25 points encompassing 2 layers around the closest training point as shown in Figure- 2(a) was created.

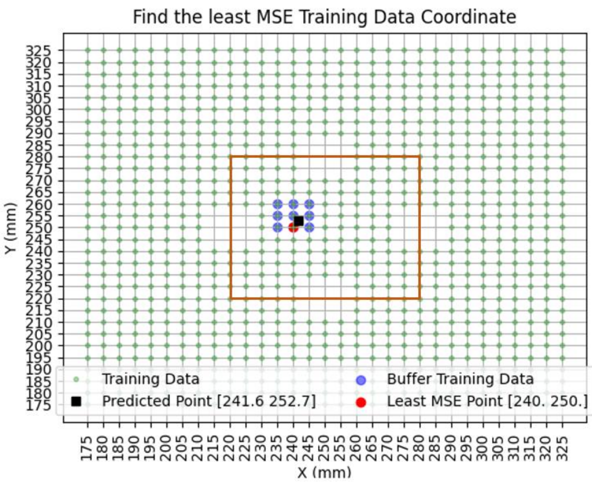

Figure 2. Buffer set surrounding predicted point located in the inner region

Taking some missing data into account 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.

Previously 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 increased accuracy. Due to this, multi-layer models with smaller kernel sizes are pursued. It was experimentally already proven that the small-size filters made the receptive field similarly efficient to the large-size filters in a parallel assignment (Alzubaidi Laith, 2021). Thus, the implemented modified VGG (Figure-2), has an input layer with kernel size 3 with convolution layers with kernel size 1 to regulate the complexity of the network. The increase in depth (addition of layers) is attributed to better performance because of the enhanced generalisation. However, it is accompanied by inevasible drawbacks - overfitting and vanishing gradient problems (Srivastava N, 2014). To avoid the problem of overfitting, for the given input size the number of layers was reduced from 16 to 11 as seen in Figure-2.

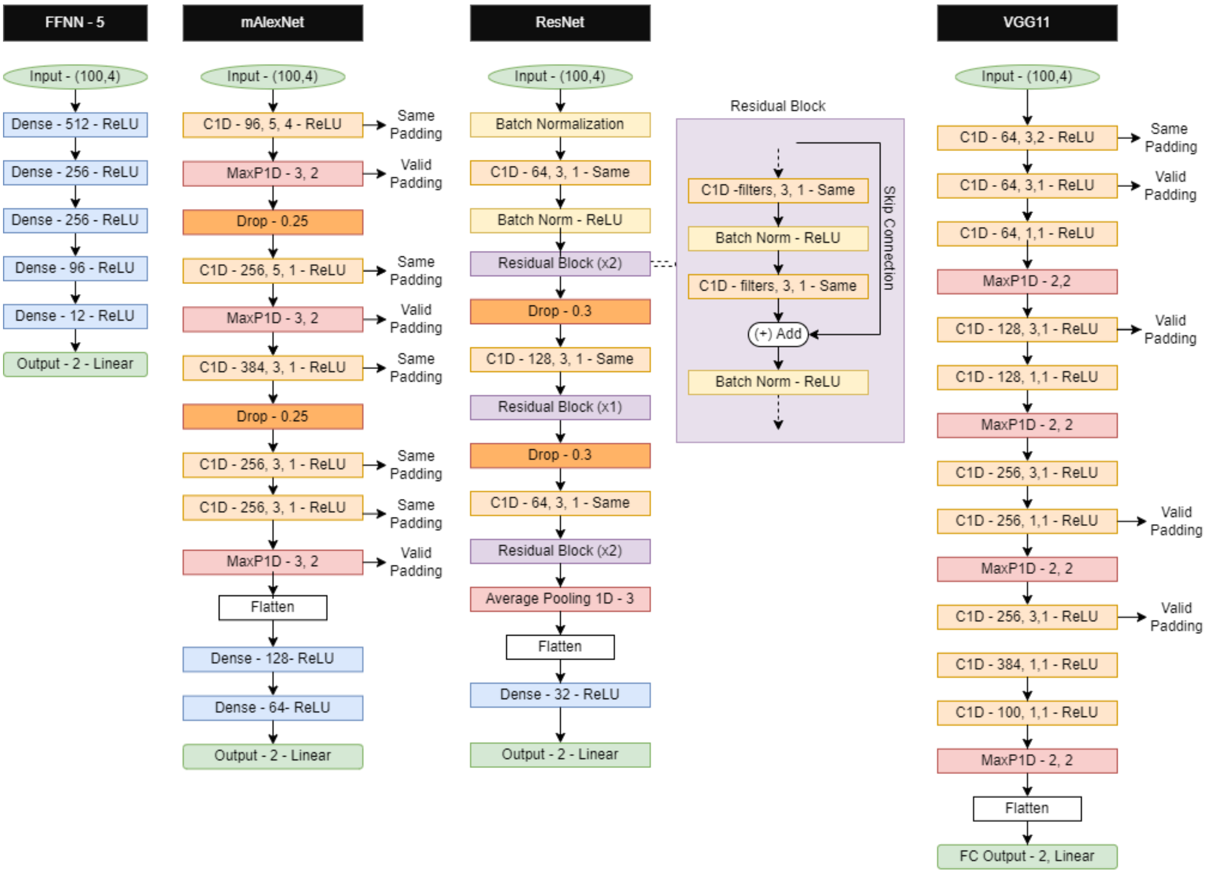

Figure 3. Network architecture implemented: FFNN, mAlexNet, ResNet and VGG

In general, 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 called the gradient of this weight and is determined by the chain rule. The local gradient increases as the gradient keep flowing backwards to the initial layers, thereby making the gradient almost constant. This leads to an increase in the run-time and stagnation of accuracy with an increase in layers, causing the vanishing gradient problem; as we build the CNN model deeper, the derivative when back-propagating to the initial layers becomes almost insignificant in value (Aqeel., 2019). To reduce this problem, ReLU was utilised as the non-saturating activation function to enhance the rate of convergence in all the implemented models (Xu B, 2015)

Further ResNet architecture is adopted to investigate the reduction in vanishing gradient through identity function. As the gradient is backpropagated, it does not decrease in value because the local gradient is 1 (Boesch, 2021). ResNet addresses this issue by the bypass pathway concept. Residual blocks, a characteristic unique to these models, create a shortcut-or-skip layer to jump over some layers in the network, reducing the computational complexity with an increased depth resulting in a shorter training time (Tsang, 2018). Typical ResNet has two or three skip layers that contain batch normalisation and nonlinearities (ReLU); the input and output from the skip layers are added in the end to pass to the next layer. Two versions with two and three skip layers respectively including a few modifications were tested on the training dataset. ResNet with two skip layers and activation following the identity addition proves to be working better, therefore discussed in this report. The implemented ResNet as seen in Figure-3 retains 5 residual blocks from the originally published ResNet architecture (Adaloglou, 2021)

RESULTS: Post-processing produced accurate predictions for most of the experimental dataset. 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.

Table 1. Comparison of the implemented Models (SVR, KNN, FFNN, mAlexNet, VGG, ResNet)

Model	Depth(layers)	MSE (Mean Squared Error) (mm)	Validation Accuracy	Training time (seconds)
SVR	NA	23.7	NA	NA
KNN	NA	23.5	NA	NA
FFNN	6	39.43	0.9568	18
mAlexNet	8	16.3	0.9838	46
VGG	11	32.1	0.9820	100
ResNet	16	46.1	0.8925	70

From table 1, it can be inferred that training and validation loss of all the AI models are under 1e-2 with FFNN and mAlexNet models reaching losses under 1e-4. Training and validation accuracy along with training time, number of trainable parameters, and average prediction error are discussed in this section for all the models. FFNN, mAlexNet, ResNet, and VGG have training accuracy of 97.7%, 97.4%, 95.4%, and 98.3% respectively with validation accuracy of 95.7%, 98.4%, 94.6%, and 94.6% respectively. FFNN took the least training time of 18 seconds followed by mAlexNet, ResNet and VGG taking 46, 70, and 100 seconds respectively. While the training time is trivial for this case, it could play a major role for AI models with a larger dataset. The number of parameters and layers plays an important role in training time along with the surrogate model and network complexity. While ResNet has a deeper network with ~400k parameters and 16 layers, a wider architecture mAlexNet performed better with ~900k parameters and 8 layers. FFNN and VGG models have ~400k and ~500k parameters with 6 and 11 layers respectively, the deeper network performing better being narrow. To compare the AI models' performance (Mean Squared Error) MSE is calculated on the predicted and ground truth of the experimental dataset. SVR and KNN regression models performed competitively to the AI architectures, notably KNN predicting better than the other. Out of all the models, mAlexNet has the best MSE of 16.3mm. While FFNN has good accuracy it proves to not predict better than VGG and ResNet on experimental data with MSE of 39.43mm, 32.1mm, and 46.1mm respectively. SVR and KNN models are close to each other with 23.7mm and 23.5mm respectively.

DISCUSSION: The list of solutions to improve the performance of the CNNs renews every day based on the dataset and the field of application. Although the depth of a neural network is crucial for its performance it also becomes rather difficult to train the network with increasing depth. Limited-dataset problems are always prone to overfitting, hence augmentation by flipping the sensor data was utilised to increase the amount of available training data, which reduces the risk of overfitting. The high accuracy throughout all the adopted models can be attributed to Augmentation.

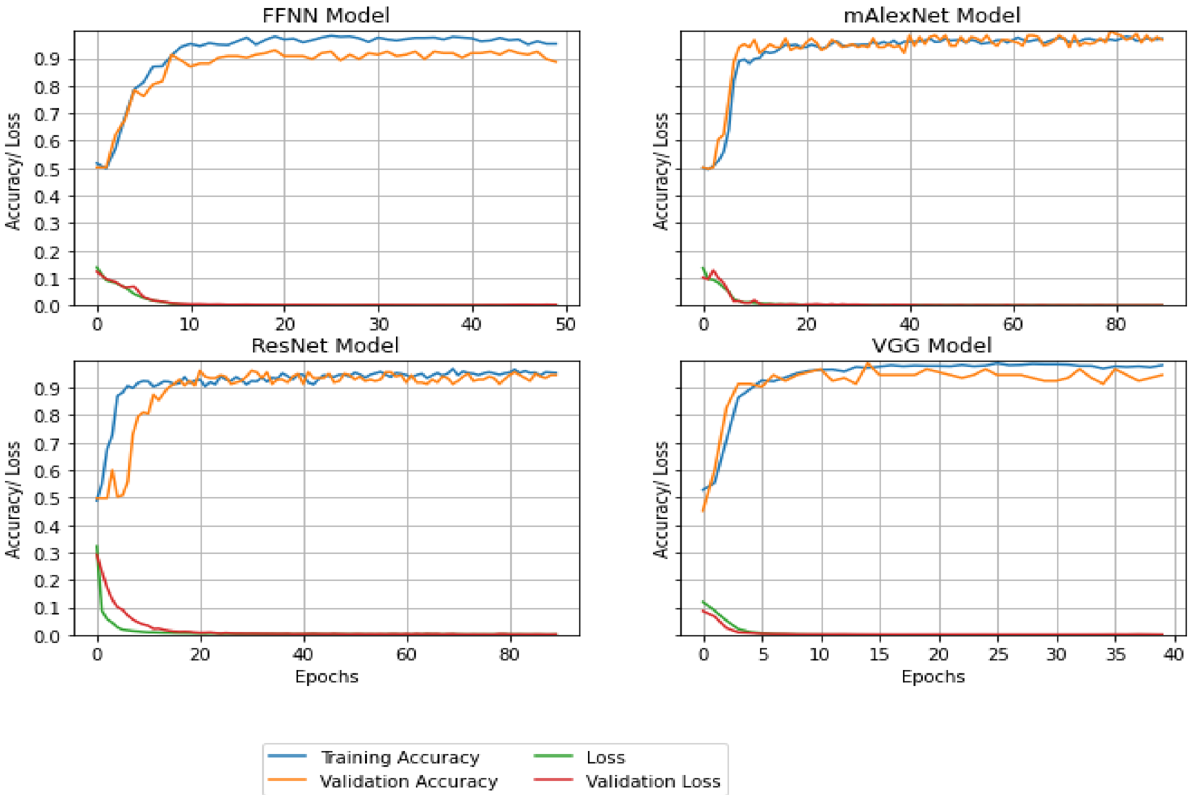

Figure 4. Epochs versus Accuracy/ Loss of (a) FFNN (b)mAlexNet (c)ResNet (d)VGG models.

Empirically, neural network models work better than machine learning algorithms as they could extract the features with no human intervention and make intelligent decisions reducing the scope of error and need for expertise on parameters (Kavlakoglu, 2020). However, it is apparent from the results that the ML algorithms (KNN and SVR) performed on the same standards as the CNNs due to the small size of the dataset. From Figure-4, it is evident that AI architectures with CNN surrogate models performed better due to the ability of the network to follow the sensor signal pattern.
It is observed that the FFNN model with the dense layers which create an input and output formed well due to a low abstraction level with a good testing accuracy (Min Lin, 2014). However, it failed to retain the same with the experimental data. ResNet and mAlexNet models were over fittings but the implementation of the L0 regularisation technique with different drop sizes from hyperparameter tuning the models showed better validation accuracy and no divergence in validation loss (Figure-4).

CONCLUSION: Through this report, the effects of post-processing on the improvement of the accuracy of impact locations' prediction and the comparison between different architectures were presented. Out of the various implemented AI models, mAlexNet worked relatively better for the given datasets. The experimental dataset provided has 3 different sets of coordinates: increased impact velocity, inner, and outer regions. The implemented mAlexNet model does not include the features and training the model with more features might yield a better outcome after post-processing. For prospects, the surrogate models can undergo ensemble methods to improve the prediction. It is one of the most common approaches. This technique combines the result of multiple weak models and produces better results. One of these methods is the boosting method (mgoel, 2018).

REFERENCES

1) Adaloglou, N. (2021, January 21st). *Best deep CNN architectures and their principles: from AlexNet to EfficientNet*. Retrieved from AI SUMMER: https://theaisummer.com/cnn-architectures/

2) Alzubaidi Laith, Z. J.-D.-S.-A. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 53-127.

3) Aqeel., A. (2019, June 7th). *Difference between AlexNet, VGGNet, ResNet, and Inception*. Retrieved from Towards Data Science: https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96

4) Boesch, G. (2021). *VGG Very Deep Convolutional Networks (VGGNet)*. Retrieved from Viso.ai: https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/

5) Kavlakoglu, E. (2020, May 27). *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?* Retrieved from IBM Blog: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

6) Luliana Tabian, H. F. (2019). A Convolutional Neural Network for Impact Detection and Characterization Lof Complex Composite Structures. *Sensors* , 1922- 4933.

7) mgoel. (2018, March 28). *Ways to improve Predictive Models*. Retrieved from ptc community: https://community.ptc.com/t5/IoT-Tech-Tips/Ways-to-Improve-Predictive-Models/td-p/542377

8) Min Lin, Q. C. (2014). Network in network (arXiv:1312.4400v3). *iclr2014*.

9) Srivastava N, H. G. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.15*, 1928-58.

10) Tsang, S.-H. (2018, December 1st). *Review: WRN - Wide Residual Networks (Image Classification)".Towards Data Science,1st December 2018.* Retrieved from Towards Data Science: https://towardsdatascience.com/review-wrns-wide-residual-networks-image-classification-d3feb3fb2004

11) Xu B, W. N. (2015). Empirical evaluation of rectified activations in convolutional network (arXiv:1505.00853). *arXiv by Cornell University*.