Classification of U.S. Air Force Flight Simulator Data Using Machine Learning and Neural Networks

Brian Mann

Background

In recent years, the U.S. Air Force has been experiencing a shortage of pilots. In 2021, the USAF was 1,900 pilots short, and in 2022, the number improved to only 1,650 (Johnson, 2022). To attack this shortfall, the USAF will need to train more and more commissioned officers. However, the training pipeline to become a pilot is far from simple. Cadets must go through basic officer training school, academic courses, as well as a plethora of simulations and training flights. Adding up all this training, it can take over 3 years for an airman to become a graded pilot in the USAF. It is thus critical for policymakers in the DoD to increase the efficiency of the pilot training pipeline.

One area the USAF is working to improve upon is flight simulation. Recently, the first ever remote instruction was given to student pilots at Laughlin AFB that connected them with an instructor stationed in JBSA-Randolph (Faske, 2024). This innovation will allow for flight instructors to maintain responsibilities at their home stations, while still providing the necessary training and guidance to students elsewhere. Similarly, the USAF has been working with the MIT AI Accelerator (AIA) program in recent years to help improve the accuracy and real-time assistance for student flight simulators (Kepner, 2021). One of their challenges is to identify and grade flight maneuvers on the simulator in real-time.

The Problem

This project aims to assist the program in their mission by accurately identifying faulty training data using both machine learning techniques, as well as a convolutional neural network (CNN). The flight simulator is based on the T-6A Texan II, which is a single-engine two-seat aircraft designed for training both USAF and USNAF pilots (USAF, 2024). Student pilots much achieve a certain level of proficiency on these simulators before flying on one of the 446 active T-6s in operation. The MIT-AIA program hopes to create an AI model that can accurately identify and grade the 30+ maneuvers that can be performed on the simulator. This project will be focusing on separating .tsv and .png files into ‘good’ and ‘bad’ categories through classification.

Dataset

The data for this project was graciously gifted by Dr. Jeremy Kepner of the MIT Lincoln Laboratory. It contains information on a set of roughly 6,500 flight simulations tracked over the course of two years. For each flight, there is a TSV and PNG file corresponding to the flight ID number contained in the file name. Each TSV file contains 10 variables amongst four categories – time (seconds after the start of simulation), position (x, y, and z variables for the position of the aircraft in meters), velocity (x, y, and z variables for the velocity of the aircraft in each direction in meters per second), and angle (heading, pitch and roll of the aircraft in degrees). The PNG files each contain a graphical plot of the aircraft's position in the x-y plane over the duration of the simulation. Additionally, there is an Excel file containing labels for flights that contained excessive taxiing, irregular stoppage, teleportation, or impossible speeds. Each of the flights has already been labeled manually with either 'good' or 'bad' by USAF personnel for use in model training.

Data Preparation

In order to extract relevant statistics from each simulated flight, each TSV file was first converted into a pandas DataFrame (DF). With the DF in place, certain features were extracted, including the total duration of the flight, ranges (for position, velocity, and angle), boolean values for teleportation and impossible speeds, and the percentage of idling for the jet (percentage of rows in the DF that had XY values unchanged). All the aggregate statistics were then placed into a new DF that contained information for every flight contained in the TSV file list. From there, the DF was cleaned to get rid of duplicate entries and fill in any missing values with zeroes.

To prepare the PNG files for entry into the CNN model, the image files were first placed into two directories for 'training' and 'testing' sets, which were then separated into the two categories ('good' and 'bad'). The training set contained an equal distribution of good and bad flight data, with roughly 1,500 images total. This was done so that the model was properly trained, mitigating the potential for it to predict 'good' for every image. The test set contained over 1,300 images, with a distribution of 85/15 good/bad images, which is roughly equal to the overall distribution across all images.

Methods

This project will be conducted in three phases – data cleaning / feature extraction, machine learning (ML) modeling, and image classification using CNNs. During data cleaning and feature extraction, data from each TSV will be taken and placed into a pandas DataFrame (DF) using a python Jupyter notebook. As each of these files contain thousands of rows with information for each fraction of a second during the simulation, it will be both unfeasible and impractical to use this raw data in ML modeling. This is why for each flight, instead of recording the raw data into a DF, certain aggregate features will be extracted, such as the total flight duration, range and maximum (of position, velocity, angle), and other values like the existence of anomalies in the data. Once this data is extracted, some initial EDA will be conducted to identify trends and help distinguish flights.

Once the data has been cleaned and prepped, it will be ready for ML modeling. First, the dataset will be split into training and test sets in an 80/20 ratio, being sure to include a roughly equal proportion of 'good' and 'bad' flights in both sets. A logistic regression and decision tree model will then be trained on the training set and then tested using the test set. There will be 5-fold cross-validation on each model, and the results of modeling will be evaluated using a confusion matrix and an accuracy report containing precision, recall, and an F-1 score.

Lastly, the PNG files will be compressed, resized, and converted into PyTorch tensors for the purpose of creating a CNN that can classify each flight into 'good' and 'bad' categories. Just as in the ML modeling phase, data will be split in an 80/20 training/testing ratio, the CNN will be trained on the training set, and then it will be graded for its accuracy in predicting the test set.

Challenges

There are numerous challenges that may arise over the course of this project. One challenge is the lack of consistency in the data. Some flights are significantly longer than others, some flights contain many different maneuvers at once, some flights contain no maneuvers, and other flights contain corrupted or incomplete data. This is the primary reason behind deriving aggregate features from the data as opposed to using raw values. Another challenge is that each of the 'good' and 'bad' labels were applied by human identification. There is a possibility that some labels have been erroneously applied. An additional challenge will be determining how best to compress and prepare the PNG files for neural network training. If the image files are too large, training time may become exponentially longer.

Exploratory Data Analysis

Before preparing the data for machine learning modeling, a general survey of the data was conducted through visual analysis. It became immediately apparent that there were clear distinctions between good and bad flight data in almost every metric extracted from the features calculated from the TSV files. In the overwhelming majority of cases, bad flight data was characterized by minimal ranges around zero in position, velocity, and angle, while good flight data generally contained a normal distribution or somewhat negatively skewed distribution. The average percentage of idling was 93% for bad flights, while it was only 10% for good flights. Only total duration was not significantly different between the two types of flight data, with the different in averages being only 90 seconds. These differences are illustrated in Figures 1 and 2. Additionally, good flights tended to have a higher percentage of instances of teleporting and high speeds (Figure 3).

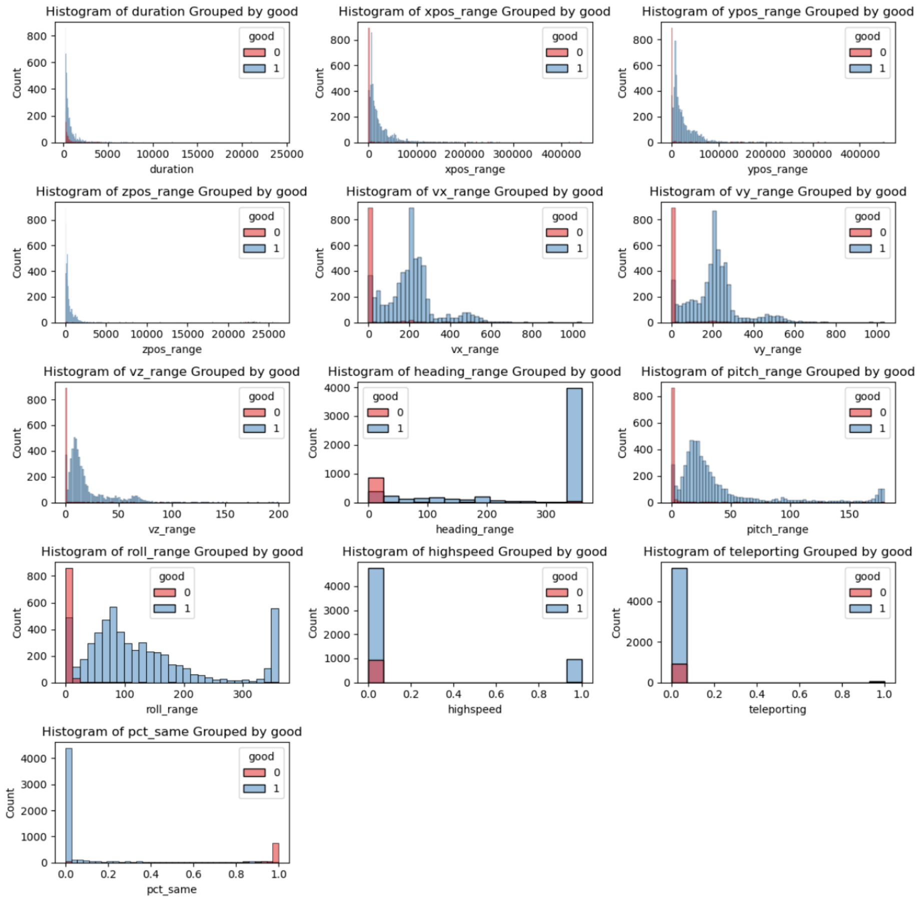


Figure 1: Histogram of Flight Simulation Features

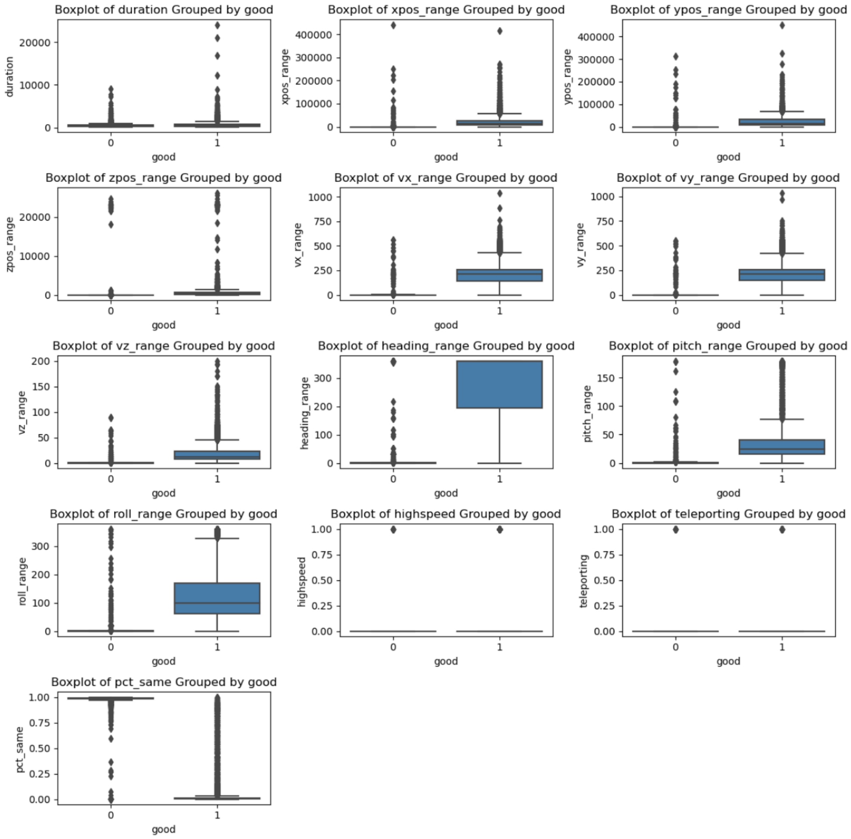


Figure 2: Boxplots of Flight Simulation Features

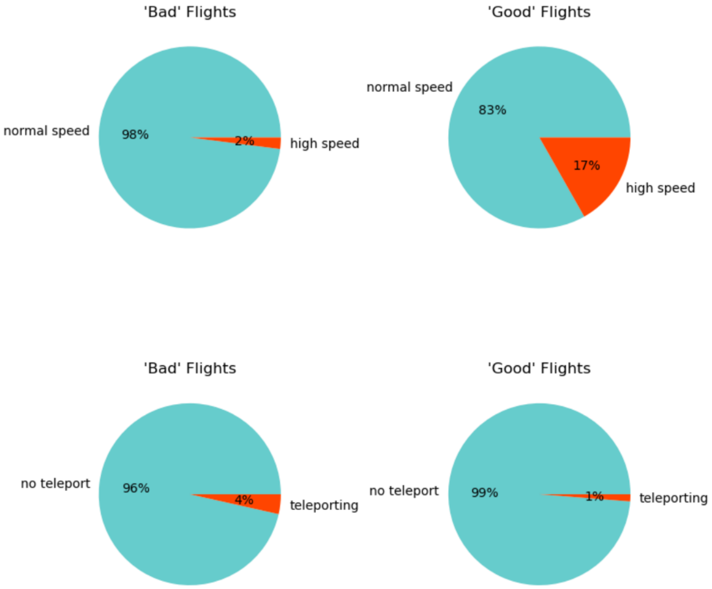


Figure 3: Ratio of Teleportation and Impossible Speeds for Good and Bad Flight Data

Lastly, a correlation heat map was produced to find out which variables had the highest absolute correlation with flight quality (Figure 4). Flight quality had the highest positive correlation with velocity ranges in the x and y direction (51-52%), as well as heading (61%). It had the greatest negative correlation with the percentage of idling, at nearly 80%. There was minimal correlation with duration, z position, and teleportation.

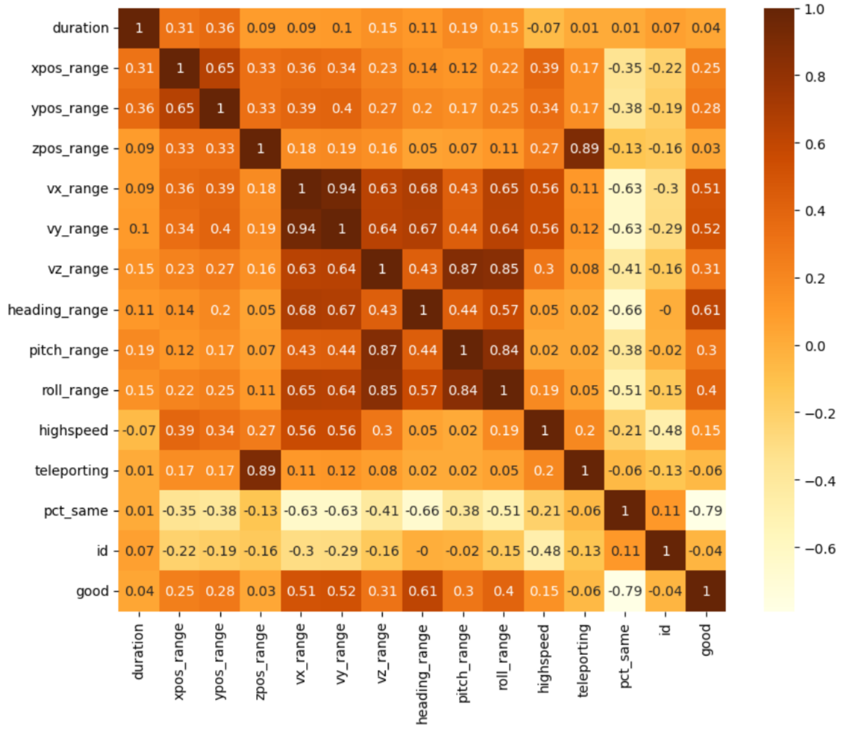
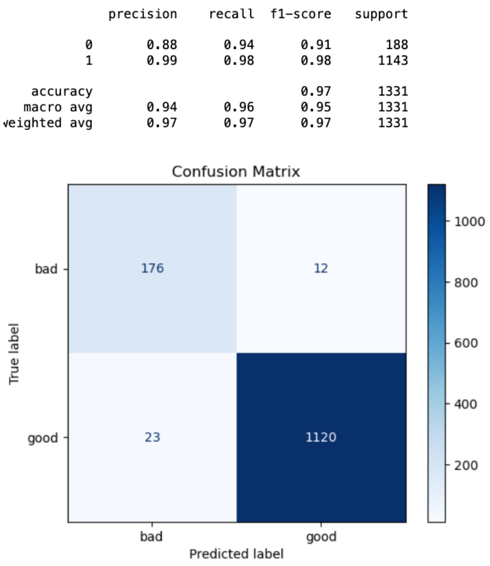


Figure 4: Heatmap of the Correlation of Flight Data Features

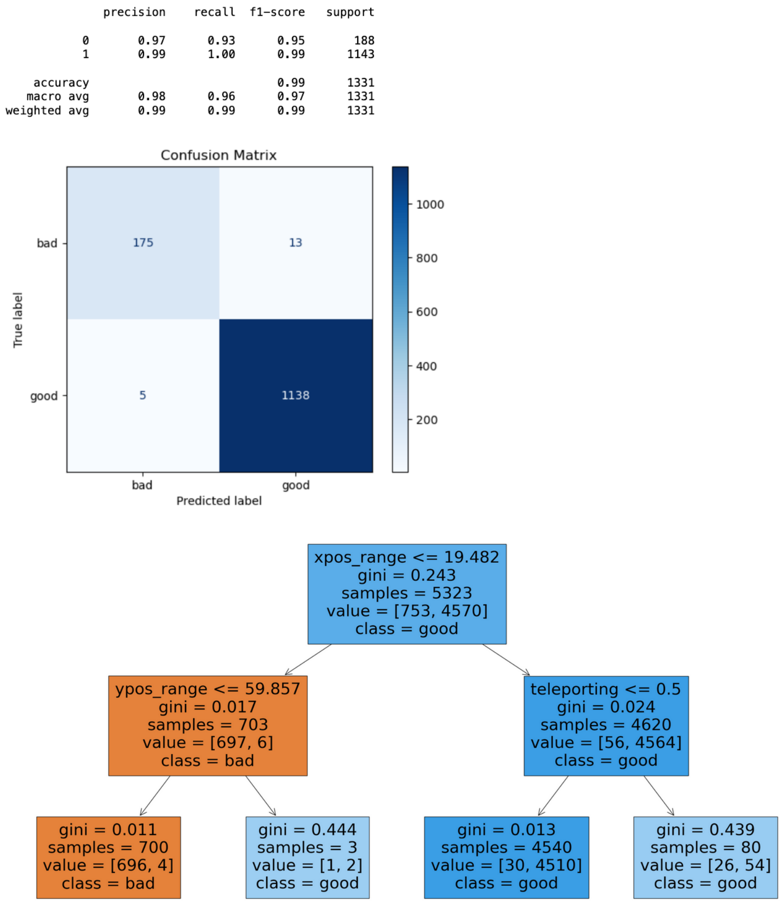
Machine Learning Modeling

To prepare the extracted TSV data for modeling, data was first split into training and testing sets in an 80/20 ratio, with the balance of good/bad quality flights being equivalent in each, relative to the overall dataset. I logistic regression model was then made after scaling the training data by means of a standard scaler. This basic binary classification model produced an overall accuracy of 97%, with F1 scores of 91% for bad flights and 98% for good flights (Figure 5). Next, a decision tree model was made using the exact same training and test sets. With a max depth of only 2, the model was able to perform with 99% accuracy, with F1 scores of 95% for bad flights and 99% for good flights (Figure 6). As long as the range of X and Y positions were below a certain threshold (20m for X and 60m for Y), the model predicted every flight as bad, while the rest were marked as good.



```
              precision    recall  f1-score   support

           0       0.88      0.94      0.91       188
           1       0.99      0.98      0.98      1143

    accuracy                           0.97      1331
   macro avg       0.94      0.96      0.95      1331
weighted avg       0.97      0.97      0.97      1331
```

Figure 5: Results of Logistic Regression Modeling



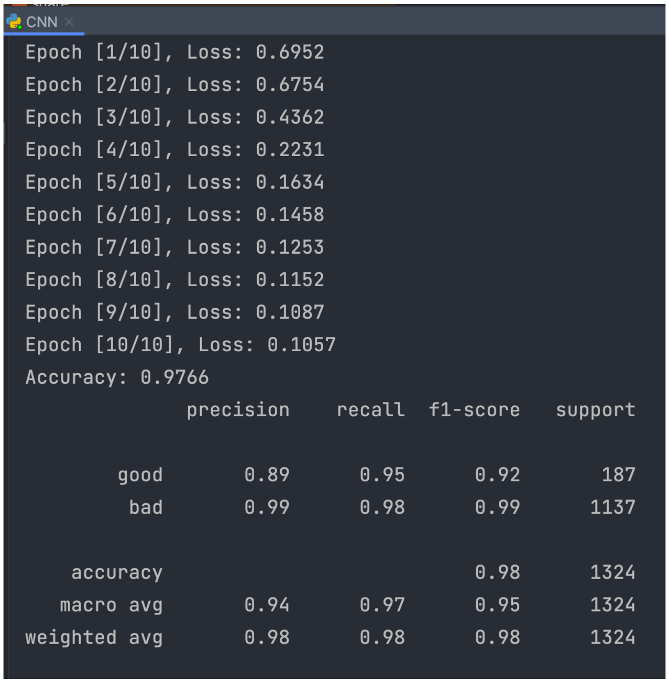
```
              precision    recall  f1-score   support

           0       0.97      0.93      0.95       188
           1       0.99      1.00      0.99      1143

    accuracy                           0.99      1331
   macro avg       0.98      0.96      0.97      1331
weighted avg       0.99      0.99      0.99      1331
```

Figure 6: Results of Decision Tree Modeling

Convolutional Neural Network Modeling

A CNN was constructed that utilized three convolutional layers. The first convolutional layer took a three-channel RGB input and produced 16 feature maps, while the subsequent layers increased the number of feature maps to 32 and then to 64, each using a kernel size of 3x3, a stride of 1, and padding of 1 to maintain spatial dimensions. After each convolution, a ReLU (rectified linear unit) activation function was applied to introduce non-linearity, followed by max pooling layers that reduced the spatial dimensions of the feature maps by half. The output after the final pooling operation was flattened into a one-dimensional vector and fed into two fully connected (FC) layers. The first FC layer reduced the dimensionality to 128 neurons, followed by another FC layer that produced the final output with two neurons, representing the two classes for flight quality (good/bad). Each image was transformed to a 128x128 Tensor, batch size was 4 images at a time, and the optimization function was Adam with a learning rate of 1e-4.

The model was able to perform with an overall accuracy of 98% after 10 epochs, with an F1 score of 92% for bad flights and 99% for good flights (Figure 7). Loss was significantly reduced across each epoch, with the most reduction in epochs 3 and 4, followed by a gradual decline in the reduction rate after epoch 5. Only 31 images were misclassified, with 2/3 being false negatives (Figure 8).



```
Epoch [1/10], Loss: 0.6952
Epoch [2/10], Loss: 0.6754
Epoch [3/10], Loss: 0.4362
Epoch [4/10], Loss: 0.2231
Epoch [5/10], Loss: 0.1634
Epoch [6/10], Loss: 0.1458
Epoch [7/10], Loss: 0.1253
Epoch [8/10], Loss: 0.1152
Epoch [9/10], Loss: 0.1087
Epoch [10/10], Loss: 0.1057
Accuracy: 0.9766
              precision    recall  f1-score   support

        good       0.89      0.95      0.92       187
         bad       0.99      0.98      0.99      1137

    accuracy                           0.98      1324
   macro avg       0.94      0.97      0.95      1324
weighted avg       0.98      0.98      0.98      1324
```

Figure 7: Results of Each Epoch and Accuracy Metrics for CNN Model

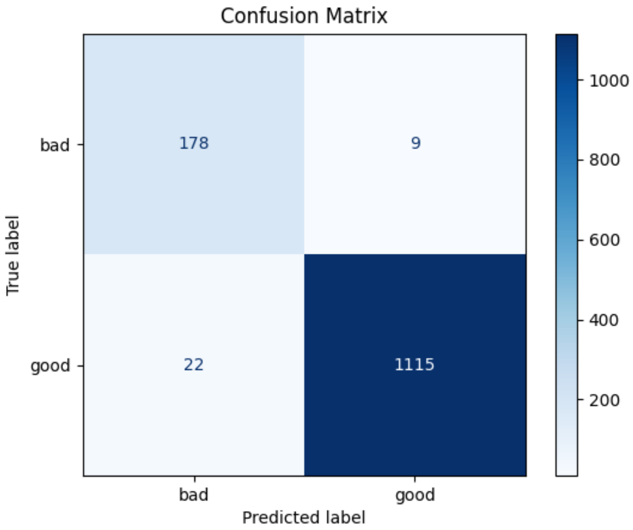


Figure 8: Confusion Matrix for CNN Model

Conclusion

In conclusion, each of the three models (logistic regression, decision tree, and CNN) performed at above 97% accuracy. The decision tree model produced slightly better results than the logistic regression model, despite only having a max depth of 2. The ranges in position in the XY plane for the aircraft over the course of flight simulation was the biggest demarcation between the quality of a simulation. The neural network was able to generate results that were slightly better than the logistic regression model, but not quite as accurate as the decision tree. Both the logistic regression model and the CNN model tended to struggle more with precision, while the decision tree had more of an issue with recall. As all the models tested in this project achieved a high level of success, we are optimistic that some of the methods developed can be implemented into real-world flight simulation classification tasks.

Assumptions and Limitations

Throughout the course of this project, there were several limitations. Due to the large amount of data contained within each TSV file, it would have been unfeasible to collect all that raw information into a single DF. Instead, it was determined that extracting aggregate features would be a good compromise, such that the data still reflected the main characteristics of each flight. We assume that the same aggregate statistics can be collected in real time during a simulation. Additionally, some modifications had to be made to the training set of images. At first, the CNN was overfitting to images of ‘good’ quality flights, due to there being significantly more good flights than bad flights in the overall dataset. The decision was made to cut the number of good flights down to the same size as the number of bad flights. This might have limited the effectiveness of training, but it significantly increased the accuracy of predictions.

Implementation Plan

It is recommended that the decision tree model and the convolutional neural network both be implemented on flight simulation data. Both have an accuracy of at least 98% on the same testing data, and both should be simple to implement on incoming flight data in real time. As a flight is taking place, if there is a prolonged period of limited changes in XY position, that range of positions can be checked against a simple if-statement to classify it into a good or bad quality flight. Similarly, images of the XY position can also be sent to the CNN for a secondary verification.

Recommendations and Future Applications

It is recommended that the results of data extraction and modeling from this project be applied to future classification problems in flight simulations. The overall goal will be to classify and grade various pilot maneuvers while student pilots are conducting their training. Similar features such as positional, speed, and heading ranges could be applied to the starts and ends of maneuvers, which would help to classify the defining characteristics of such a maneuver. These kinds of techniques may also one day be applied to actual jets, improving flight quality and operational effectiveness of the USAF in the future.

Ethical Considerations

A major consideration in this project is operation security (OPSEC) for the U.S. Air Force. The USAF would not like the intricacies of flight simulation data to be accessed and exploited by a foreign adversary, even if it is only for initial pilot training. This is why all the positional data for each flight has been normalized to sanitize the specific capabilities of the simulator. Additionally, even though this data is unclassified, special permission had to be given to access it. Due to this, precautions will be taken to not publish any of the raw data to any public repositories such as GitHub. However, the code and results used in this project will be made available on GitHub, with permission from the MIT-AIA team. As this project will not be aiming to classify any specific pilot maneuvers, that will not be a factor under consideration.

References

Faske, B. (2024). *Pilot training innovation: First successful remote simulator training*. Air Education and Training Command. https://www.aetc.af.mil/News/Article-Display/Article/3661214/pilot-training-innovation-first-successful-remote-simulator-training/

Johnson, K. (2022). *U.S. Air Force is Short 1,650 Pilots, Report* Says. Flying Magazine. https://www.flyingmag.com/u-s-air-force-is-short-1650-pilots-report-says/

Kepner, J. (2021). *AIA Maneuver Identification*. MIT AI Accelerator. https://maneuver-id.mit.edu/

USAF. (2024). *Our Mission*. DAF AI Accelerator. https://www.aiaccelerator.af.mil/About-Us/

USAF. (2024). *T-6A Texan II*. T-6A Fact Sheet. https://www.af.mil/About-Us/Fact-Sheets/Display/Article/104548/t-6a-texan-ii/