Deep Learning Boiling Detection
Final Report

Cal Poly: Alexander Arrieta, Nathan Hill, Soren Paetau
NASA: Michael Khasin, Matteo Corbetta
Faculty Advisors: Hunter Glanz, Jonathan Ventura

Abstract

Boiling fuel in rockets raises many potential dangers. It would be greatly beneficial to determine the conditions inside of the fuel tanks in real time to mitigate the risks associated with boiling fuel. Boiling typically emits distinct audio signals that can be used to identify its exact stages and characteristics. However, detecting localized boiling inside a fuel tank presents numerous challenges due to the inability to gather data directly from inside the tank, instead relying on external sensors. We analyzed accelerometer data from an isolated experiment using several different methods to determine the most efficient way to categorize the data. The experimental setup varied to capture different possible boiling conditions within the tank and external noise. We found that there were several viable methods of change detection given the experimental data. However, due to a lack of variety in the available experimental data it is unclear if simpler methodologies will be effective in real life scenarios given the amount of background noise that could be expected and the wide range of possible scenarios. Still, our methodology has shown that it is possible to classify boiling within a fuel tank with over 90% accuracy despite the loss in data quality using external sensors.

Background

The primary objective of this research is to explore whether there are specific characteristics of boiling that can be detected by accelerometers in high noise environments. Specifically, we are interested in whether boiling within a fuel tank can be detected by external accelerometers on a rocket. The detection of boiling is essential to mitigate the damages as quickly as possible, against the risk of catastrophic consequences. This primary objective can be split into three smaller parts. First, we want to know if accelerometers can detect boiling in general, like audio data has been able to do. Secondly, it should be reasonable for the model to classify fast enough for real time predictions. Finally, we want to see if the model is generalizable to different levels of boiling initiated in different locations.

Our initial work builds upon previous studies (Sinha et al.) that used hydrophones and a Convolutional Neural Net (CNN) to classify boiling regimes. These studies were highly successful in their goal with accuracy metrics larger than 99%. We set out to discover if substituting hydrophones with accelerometers could achieve the same successes, and whether the model would be generalizable to more extreme conditions.

Data Cleaning

Before we do any analysis of the data, we must label and organize it. We combine similar data files into a single file containing data from multiple runs. When the sampling rate is at 10kHz, we take care to only keep data points in multiples of 1000 from each run so that there are never any splits that contain data from two different experiments. Similarly, 50kHz data is kept in multiples of 5000 points.

The data delivered by NASA can be combined into two major groups:

The first contains data from experiments conducted on February 29 and March 5, 2024. This group of data is composed of five files containing background noise only, three files of stationary boiling at various temperatures, and four files with calm to boiling data.

The background noise files have lengths of 10 seconds, 9.7 seconds, 9.2 seconds, 10 seconds, and 9.8 seconds. Altogether this provides us with 48.7 seconds of data for each of the two accelerometers, or 97.4 seconds combined.

The stationary boiling files have lengths of 10 seconds, 6 seconds, and 9.9 seconds. The heater was set at different temperatures for each of these runs. The temperature measurements were recorded by a thermocouple a set distance away from the heater, which provided temperatures of 85, 93, and 98 degrees Fahrenheit respectively. Once combined, each of the accelerometers have a total of 25.9 seconds of data, or 51.8 total.

The calm-to-boiling data consisted of experiments that began at a set background temperature, and the heater was slowly heated to induce boiling during data collection. There is no true point where boiling begins, thus we had to estimate where to separate the two classes based on visual estimates. The files contain 7.9, 5.6, 12, and 20 seconds of data with starting temperatures of 65, 80, 77, and 86 degrees Fahrenheit. Overall, there are about 35.4 seconds of boiling data for each accelerometer and 10 seconds of background noise.
The total length of data we have from these experiments is 240 seconds, composed of 122.6 seconds of boiling and 117.4 seconds of background noise.

All of the data was categorized and labeled as either boiling or not boiling. This labeling was provided for each data point independently. This labeling process was initially carried out using solely the stated information about the audio files and a visual inspection of the data as a waveform and spectrogram. The further data about the experiments provided in the second set was also factored into that data labeling. However, it should be noted that the accuracy of the labeling may not be 100%, and that different definitions of what it means for the fluid to be boiling could affect how the data is labeled. While this poses minimal risk in terms of providing a poorly trained or tested model, it may affect some of the statistics and information gathered from the full procedure. There is no necessarily correct answer to these concerns, but for future work there must be a consistent standard in place for labeling data, and we present all results in this report relative to our definition of boiling. We focused on bubble release from the nucleation site as our definition of boiling.

Also note that currently our experiments involve boiling water in a controlled environment. It is unclear how boiling fuel in microgravity conditions will affect the performance of accelerometer-based detection and how the inclusion of rocket noise will interfere with model performance.

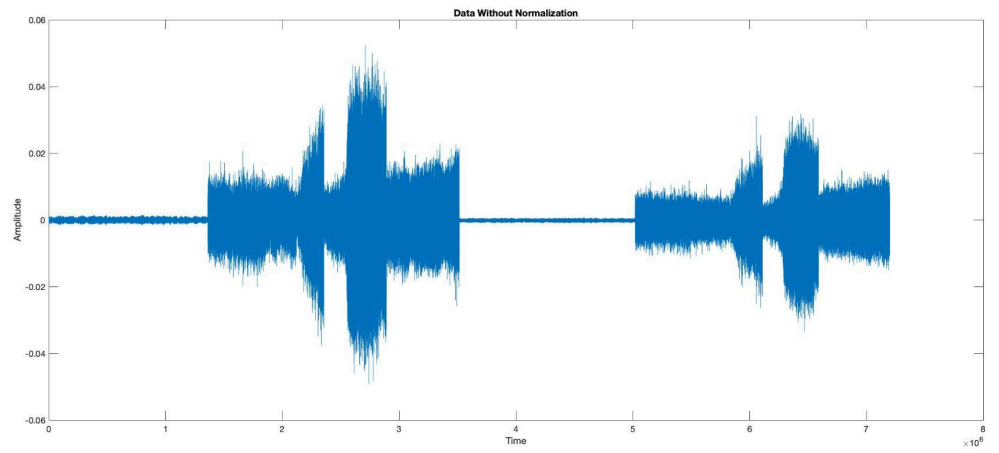
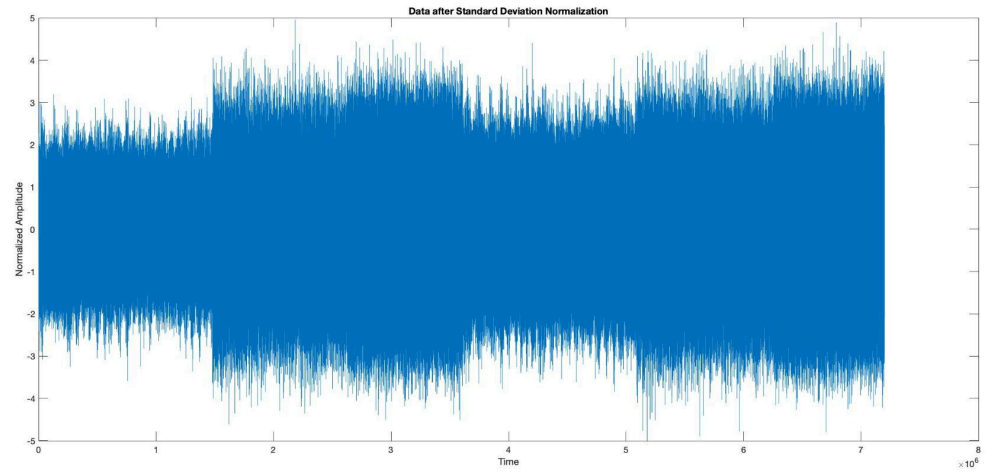
Data Analysis

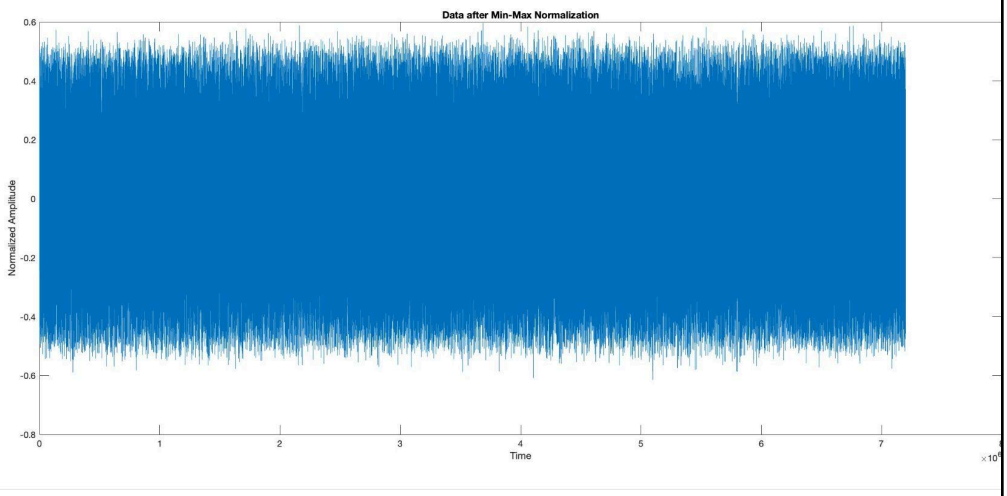
For our data analysis, we keep data from each of the two accelerometers separate, but feed them into the same model. The full data file is read in and split into 0.1 second chunks. At the 50kHz frequency, we have 2400 chunks of 5000 data points for the first set of data.

Next, we partition the data into training, validation, and testing sets using a 60/20/20 split. The data is randomly partitioned into the three groups, and we expect to see roughly the same distribution of data classifications each time.

The data is normalized to ensure that amplitude alone does not separate boiling data from background noise. We attempted to normalize the data in a variety of ways, but in the end we found that a min-max normalization is the best for ensuring even amplitude across all of the data. This normalization is performed by dividing each 0.1 second chunk by its range.

Figure 1: Comparing Normalization Techniques
The min-max normalization appears better suited for the accelerometer data than other methods

Spectrum of the full dataset without normalization	
Spectrum of the full dataset normalized by standard deviation	

Spectrum of the full dataset with min-max normalization	

Visually, the min-max normalization performed better than other normalization techniques at evening out the amplitude across all chunks. The superior performance in evening out amplitude is also demonstrated quantitatively by dividing the standard deviation of the maximum amplitudes by the range of the maximum amplitudes. We want to see lower values, which indicate reduced variation among the maximum amplitudes.

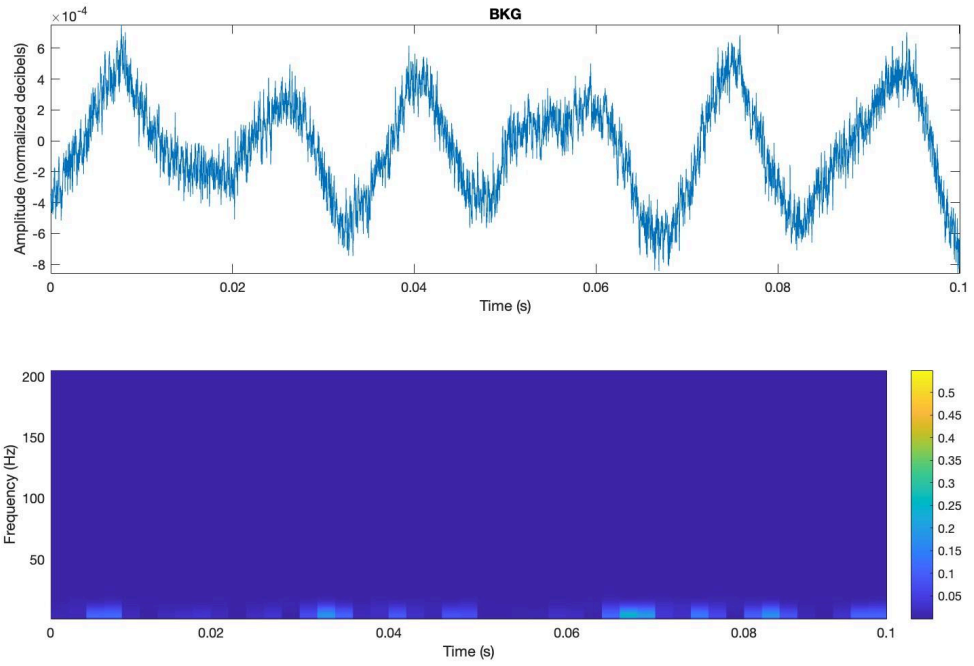
Figure 2: Variances of Maximum Amplitudes
The Min-Max Normalization reduces the variances among maximum amplitudes the best after accounting for the range

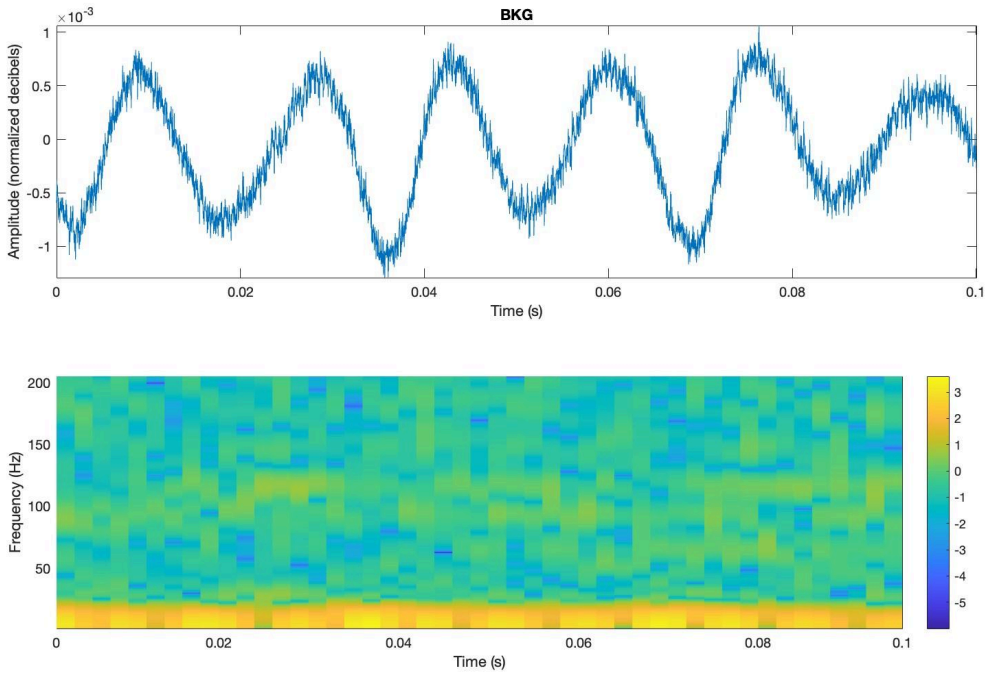
	Standard Deviation of Maximum Amplitudes / Range of Maximum Amplitudes
No Normalization	**0.1963**
Standard Deviation Normalization	**0.1583**
Min-Max Normalization	**0.1363**

We then utilize a fast Fourier transform to alter each segment of raw data so the frequency information can be visualized. This is performed through the audioFeatureExtractor function, which outputs a linear spectrum. The data is now in a 2 dimension format with the axes being frequency and time. For plotting we have the frequency along the y-axis and time along the x-axis. The actual measurement at each time-frequency pair is one of intensity. The higher the number the higher the intensity (or relative intensity for normalized data). Frequency is measured in 1 Hz units, so for capturing the 1 to 100 Hz range we have 100 discrete values for the y-axis. Time is measured in discrete chunks as well, depending on the sampling rate of the data. Thus, the number of discrete values along the x-axis depends on the sampling rate and length (in

seconds) of the data chunk. The final transformation also happens here; we put the data through a logarithmic transformation to accentuate the highlights in the spectrogram. This transformation has no impact on model performance but is extremely useful for identifying features visually.

Figure 3: Motivation of a logarithmic transformation
Dividing the data by a factor of the window size and taking the log makes the spectrogram features more visible without sacrificing data quality

Example without log transform	

Example with log transform	

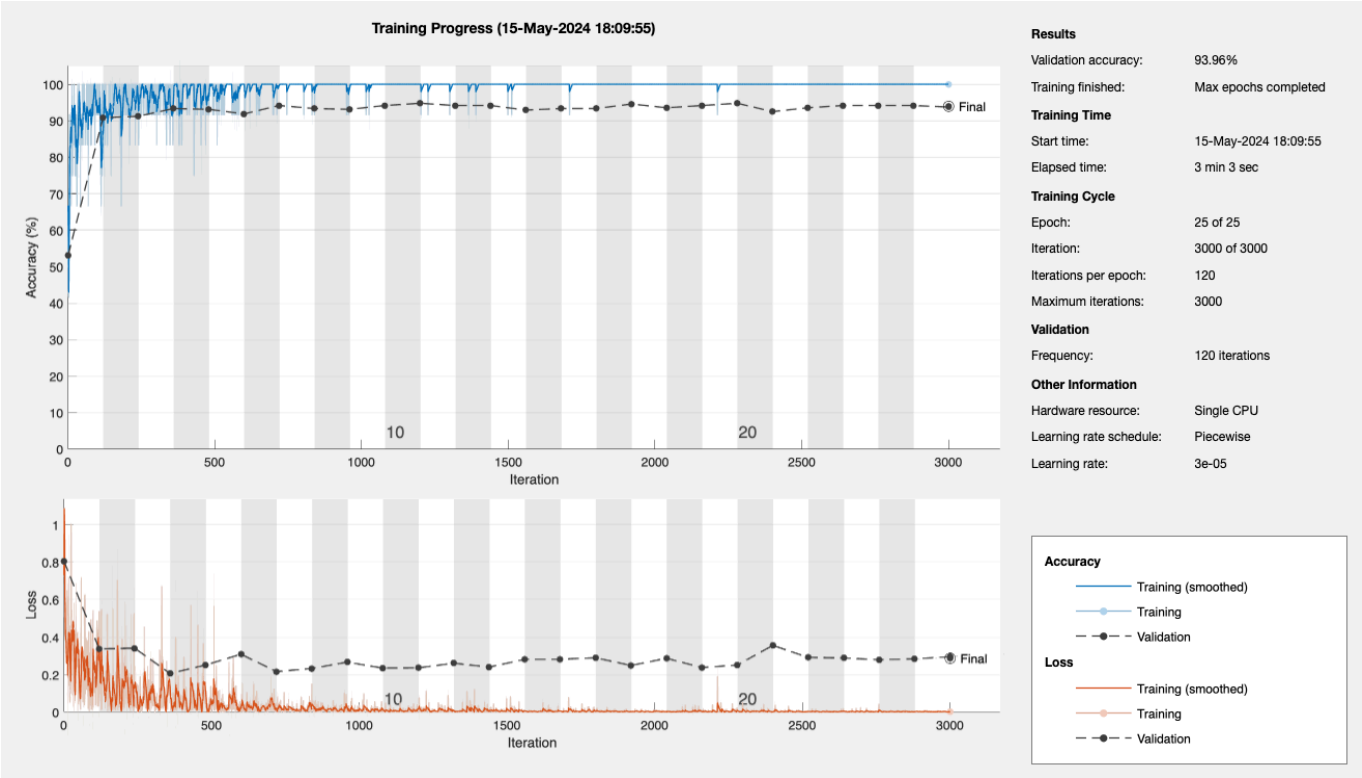
At this stage, the data is ready to be fit to a CNN. We define the parameters and architecture of the model before inputting the data.

Model Description

Our primary model is a Convolutional Neural Network (CNN) as recommended by the previous work of Sinha et al. and other authors. This particular model architecture is designed to do dimension reduction on the data. Each data chunk (the .1 second chunks) is a large collection of actual data points after processing by the FFT. The images of the spectrograms shown, which is the input of the CNN, can have nearly 10,000 individual numbers. The exact number of data points is the number of frequencies shown times the amount of time chunks the interval was split into. It can be thought of as how many pixels the image has. The CNN will combine information for nearby pixels and replace them all with the new number. For example if we have a 2 by 2 set of pixels we may combine them all into one pixel using the mean of the 4 numbers. Thus we have ¼ of the original size of data. This allows for a more conventional neural network to have a much smaller input space and be more able to effectively work with the data given. We present other, simpler, models in the results section as a benchmark to test the CNN effectiveness. A full description of the exact model architecture can be found in Appendix A.

Results

Figure 3: Model Output
The CNN is trained throughout 25 epochs, reaching a final validation accuracy of 93.96%



With the test/train split described above, the CNN model achieved a validation accuracy of 93.96% using the normalized data with the FFT performed. Furthermore, there was only one observation out of the 224 true boiling observations that was misclassified as background noise.

Figure 4: Confusion Matrix of Results
The confusion matrix reveals that the most substantial source of error was background noise being incorrectly classified as boiling.

		Predicted Class	
		BKG	**BOIL**
True Class	**BKG**	**228**	**28**
	BOIL	**1**	**223**

In addition to using the CNN model we also examined and tested other classification techniques to provide a benchmark for the CNN. This way we can better understand how effective the CNN is for this task and analyze what potential improvements we could make to our modeling process. The first model examined was a simple one based on categorizing the data point based only on the mean intensity of the data across all frequencies and across the full time. This acts in a not entirely dissimilar fashion to the CNN as it tries to reduce the amount of input data, and then separate out the reduced input into the different categories. All observations with a mean intensity over a specified threshold were classified as boiling and everything else was classified as background. Below is the confusion matrix for this mean thresholding technique on the raw data. The accuracy is as high as the accuracy achieved by the CNN, implying that the CNN could likely just be learning to discern the means. Thus, using the raw data without any normalization technique is unsuitable to prove the model's capabilities.

Figure 5: Confusion Matrix of Mean Thresholding
The confusion matrix reveals that splitting on the mean presents a viable strategy for the given data

		Predicted Class	
		BKG	**BOIL**
True Class	**BKG**	**419**	**56**
	BOIL	**1**	**305**

However, when the data is normalized using the min-max normalization discussed above, attempting to separate the data by mean intensity fails to separate the data at all. The maximum accuracy it archives is simply classifying everything as background due to the data imbalance. Unfortunately, using the median as the separator proves to be quite effective even on the normalized data. As seen in the following confusion matrix, although the rates are slightly worse, overall the drop in accuracy is only about 8% compared to the previous technique on the raw data.

Figure 6: Confusion Matrix of Median Thresholding
The confusion matrix reveals that splitting on the median presents a viable strategy for the standardized data

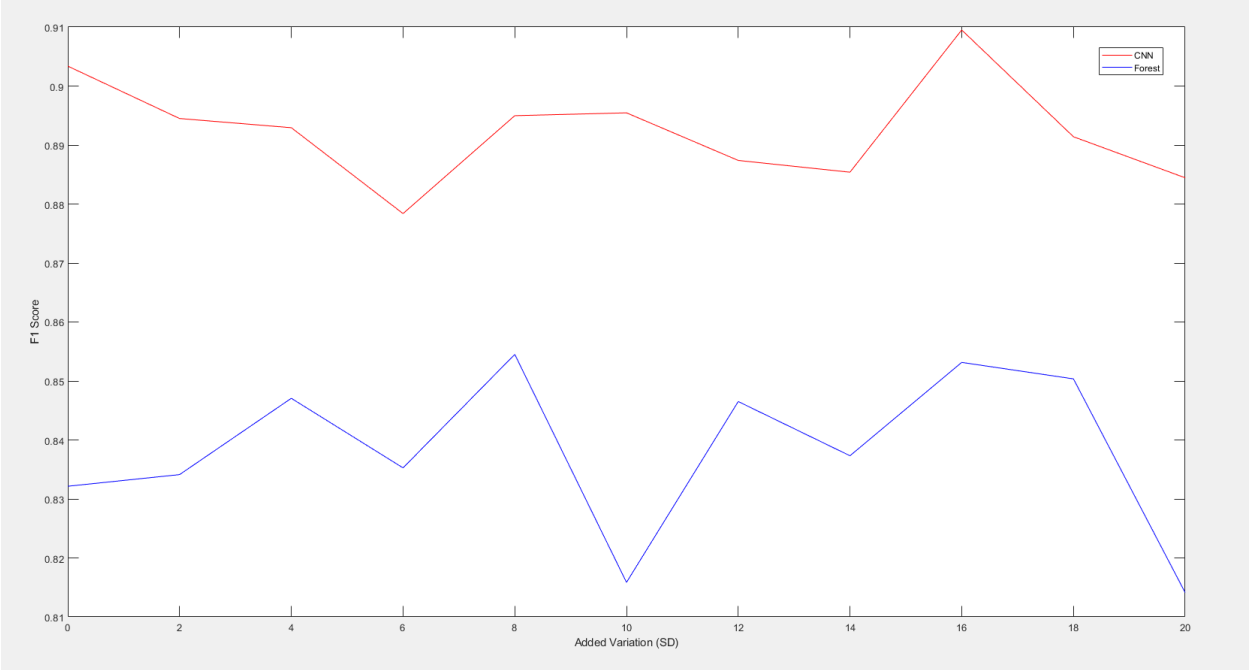
		Predicted Class	
		BKG	**BOIL**
True Class	**BKG**	**446**	**22**
	BOIL	**92**	**221**

Overall this indicates that the training data we had was simply so separable that any basic methodology for separating them was going to be quite effective. This calls into question the need or efficacy of the CNN. If we believe that the simpler models will be ineffective in real life scenarios, then the CNN could also prove to have large accuracy decay when exposed to different data.

Data Augmentation

Another approach we tried to better understand the abilities of the CNN model was to see how it responded when artificial noise was added to the normalized data. Our version of this test involved adding gaussian noise specifically. If the CNN model continues to be accurate as noise is added, this provides good evidence that the noise present in real life data will not prove to be problematic. This noise robustness test was also carried out on a secondary model to again provide a benchmark. This time a random forest classifier was used. A random forest classifier uses a technique similar to the thresholding explained in the previous section. The random forest will take in all the values it gets as input and at each step split the data into two groups depending on one of their values. For this random forest the input data was the mean, median, and variance of the intensities in the data chunk. In figure 8 we see the results of this experiment. Overall both methods maintained high accuracy as additional noise was added to the data. We measured the added noise in terms of added number of standard deviations to the variance of the data. With both methods performing well despite the added noise we see two facts about the data. The first is that generally the data has quite low variance, and the noise intensity peaks caused by boiling generate large outliers that are not fully smoothed by the normalization process. It also informs us that there are potentially simpler solutions than the CNN to solve the classification problem. The random forest was able to achieve a F1 score almost on par with the CNN with only 3 inputs, instead of the thousands the CNN had. Further testing of this type could be run, both in terms of amount of noise added and type of noise added. If added noise is targeted at a specific frequency or added in a specific pattern it could affect the models far more than gaussian noise.

Figure 8: F1 Scores for models with noised data
The models were trained and tested on the dataset with added noise, and the F1 score with boiling as the target category was calculated



Additional Datasets

The dataset containing background noise, stationary boiling, and calm to boiling observations is the primary dataset which we use to train and test our model. We are also interested in the classifications of different test sets. These sets can provide important information about the generalizability of the model and its ability to detect boiling in more varied scenarios. The sampling rate of additional data is 10kHz, thus the training data needed to be downsampled from 50kHz to match the new information. There are also temperature measurements, visual observations, and videos corresponding to these experiments.

Firstly, sets of additional data were collected on April 2nd and 3rd. Both datasets contain five files, each with five seconds of data from two different accelerometers. This data highlights the ambiguity surrounding the onset of boiling, as bubbles form on the heater without releasing or forming strong convection. Bubble formation without strong boiling is potentially a large issue with the model, as it could rely on certain characteristics of boiling to detect vibrations that are not present in the bubble formation phase.

Figure 9: Blind Classification of April Data
After training the model on the original dataset, the model was fit to new data with a smaller difference between the boiling and non-boiling runs

Date of Experiment	% Classified as Boiling	Actual Classification
4/2	2.00%	Tiny Bubbles, Some Freeing
4/2	2.40%	Tiny Bubbles, Some Freeing
4/2	1.78%	Tiny Bubbles, Some Freeing
4/2	1.43%	Tiny Bubbles, Some Freeing
4/2	11.10%	Boiling Around Heater
4/3	0.00%	Very Few Bubbles Freeing
4/3	0.00%	Some Bubbles Freeing
4/3	0.00%	Some Bubbles Freeing
4/3	5.71%	Boiling Around Heater
4/3	9.32%	Boiling Around Heater

The dismal results from this testing data support the idea that initial stages of boiling have different characteristics than more advanced stages, and the model does not pick up on very light boiling. This is problematic because the purpose of the model is to identify boiling at its first onset in order to mitigate risks associated with it. The model should somehow be trained with more data from the very first stages of boiling so it can recognize it.

The other sets of additional data are from May 13th and 23rd. The set from May 13th contains two files, while the May 23rd run contains four. Each file holds 10 seconds of data from two accelerometers as before. A primary difference in this data is the location of the heater, which is now on the external side of the tank instead of submerged within. This creates a hot spot, localizing boiling in a specific area.

Figure 10: Blind Classification of May Data
After training the model on the original dataset, the model was fit to new data from a different experimental setup

Date of Experiment	% Classified as Boiling	Actual Classification
5/13	98.50%	No Boiling
5/13	98.80%	Boiling at Hot Spot
5/23	0.00%	No Boiling
5/23	40.00%	Boiling at Hot Spot
5/23	36.78%	Boiling at Hot Spot
5/23	33.36%	Boiling at Hot Spot

This new experimental setup yielded substantially worse results than previous test sets. In the first experimental run, the model classified nearly everything as boiling, while it was reluctant to classify much boiling at all in the other experiment. The poor accuracy metrics suggest a lack of generalizability in our model.

Feature Extraction

An analysis of how the model worked was performed to understand what features are the most important in the classification process. This was done by analyzing how the model responded to targeted data augmentation. First the original model trained on the unaltered and normalized data set was fed altered images to classify. The alteration consisted of setting 5 Hz frequency bands to a maximum or minimum value to see if the model was using the bands values in the classification. For certain bands we saw great changes in classifications and for others there were minimal changes. Two representative bands are shown in figures 11 and 12.

Figure 11: Confusion Matrix for altering bands 150 to 155
The confusion matrix shows that these bands being high intensity seems to typically correspond to background noise over boiling

		Predicted Class	
		BKG	**BOIL**
True Class	**BKG**	**304**	**0**
	BOIL	**149**	**68**

Figure 12: Confusion Matrix for altering bands 60 to 65
The confusion matrix reveals that overall these bands seem to be ignored by the CNN in classifying the data

		Predicted Class	
		BKG	**BOIL**
True Class	**BKG**	**270**	**34**
	BOIL	**6**	**211**

What we saw is that only the 1 to 10, 105 to 115, and 145 to 155 Hz ranges had significant impacts on the models classification.

The follow-up step was to see if the model could gain enough information from only a subset of data to determine if it was boiling. What was found is that the model could generally do the classification task with any 10 Hz band, regardless of its effects in the earlier test. While not every 10 Hz band was tested, a representative sample of 1 to 10, 60 to 70, 100 to 110, 150 to 160, and 190 to 200 were tested and figure 13 shows the worst performing model. This model still has over 70% accuracy, and does not show a strong bias to misclassifying boiling or background data.

Figure 13: Confusion Matrix for using bands 190 to 200
Despite being the worst performing model, this model still achieved relatively high accuracy

		Predicted Class	
		BKG	**BOIL**
True Class	**BKG**	**244**	**69**
	BOIL	**87**	**121**

With no particular band showing much higher performance than the others this clearly indicates that boiling causes noise at every frequency. Moreover this noise is loud enough to be distinguishable from just background noise. However, this does not indicate that every band gets equally as loud, just that each band gets loud enough to tell the liquid is boiling. Future analysis could indicate with more certainty which frequency bands are best to focus on for the classification.

Discussion

We see the CNN created by Sinha et al. functions well for the new data provided by NASA, as long as the training and testing sets come from similar experimental setups. We do not have conclusive evidence that the model will be robust enough to handle data collected from real rockets. This is due to the data collected by NASA having very specific features that allows simple models to have high performance. Just doing a simple split on the average strength of the signal after processing with the FFT provides over 90% accuracy in boiling detection. The normalization techniques applied to the data removed some of the ability to easily classify using overly simple modeling techniques, but does not replicate the large possible variance in noise that may occur in a real rocket. The process of adding artificial Gaussian noise to the data does simulate the randomness of real noise better, but leaves the signal too clear to have any noticeable effects before the noise was stronger than the acoustic signal altogether. The model performing well under both circumstances does lend evidence to its robustness for real life scenarios, but that remains unprovable without a larger sample of data. It is also unclear if there are distinct data features that are critical for the classification task. Still, we recommend continuing to study this procedure as it shows great potential to be useful in doing real-time analysis of fuel tanks despite the limited information available.

Figure 11: Classification Times
The times for the model to classify 1, 5, 10, and 100 observations suggest that real-time analysis is viable.

Number of Observations	Time to Classify (seconds)
1	0.0020
5	0.0038
10	0.0094
100	0.0315

In conclusion, accelerometers alone provide enough information about a tank to understand its internal state, and more sensing equipment is unneeded for the current situation. It remains to be seen if the intensity of the noise generated by a rocket will substantially influence the signal.

Future Work

This project is overall in the early stages of development. With the CNN architecture and data pipeline confirmed to work with the accelerometer data the clear question left unanswered is whether the model will work in real life scenarios. This problem needs to be addressed from both a data perspective and a model perspective. To cover data concerns the most clear course of action is to gather more data from a wide range of scenarios that more closely replicate real life conditions. We particularly believe more training with data near the point of initial boiling would be beneficial. There is also the possibility of performing more data augmentation techniques. The data augmentation done within this paper is relatively small in scope from what is possible. By applying different noise patterns or combining or layering data there is a great wealth of new data that could be generated with minimal expense and wait time. When looking at future work from the model perspective the greatest question lies in what features are important for the model to be accurate. Due to the nature of our feature extraction technique and the data collected thus far, we did not find any particularly useful features within the data. However, we focused on our trained CNN model for this experiment. Another route is to do a more statistical analysis on the data using techniques like principal components analysis to derive a mathematical measure of which features have the most separation between background and boiling noise. Still, it is worth noting that at best many techniques tell us which features are “most” important relative to other features. It is still entirely possible that many of the features provide reasonable separation between the two classes, and that the best features for the current data will not be the best for future data. Even so, it is worth pursuing this type of analysis to better understand the data. Once this is complete and if additional data proves to be troublesome to the model, then it may be worth looking into other model architectures. We have not seen a need to do this at the current state of the project because as shown in the paper most models would do relatively well on this task, providing little insight into which is best.

References

Sinha, K. N. R. , Kumar, V. , Kumar, N. , Thakur, A. , and Raj, R, “Deep Learning the Sound of Boiling for Advance Prediction of Boiling Crisis,” Cell Rep. Phys. Sci. 2021, 2(3), p. 100382.

L.-T. Zhu, X.-Z. Chen, B. Ouyang, W.-C. Yan, H. Lei, Z. Chen, Z.-H. Luo, Review of machine learning for hydrodynamics, transport, and reactions in multiphase flows and reactors, Ind. Eng. Chem. Res. 2022, 61, 28, 9901–9949

Hughes, M. T.; Kini, G.; Garimella, S. Status, Challenges, and potential for machine learning in understanding and applying heat transfer phenomena. J. Heat Transfer. 2021, 143 (12), 120802

Appendix A - CNN model architecture

Layer 1: Input image layer
Layer 2: 2D convolutional layer with 12 3x3 filter with “same” padding
Layer 3: Batch Normalization
Layer 4: ReLU
Layer 5: Max pooling layer with a 3x3 pool size, a 2x2 stride, and with “same” padding
Layer 6: 2D convolutional layer with 24 3x3 filter with “same” padding
Layer 7: Batch Normalization
Layer 8: ReLU
Layer 9: Max pooling layer with a 3x3 pool size, a 2x2 stride, and with “same” padding
Layer 10: 2D convolutional layer with 48 3x3 filter with “same” padding
Layer 11: Batch Normalization
Layer 12: ReLU
Layer 13: Max pooling layer with a 3x3 pool size, a 2x2 stride, and with “same” padding
Layer 10: 2D convolutional layer with 48 3x3 filter with “same” padding
Layer 11: Batch Normalization
Layer 12: ReLU
Layer 10: 2D convolutional layer with 48 3x3 filter with “same” padding
Layer 11: Batch Normalization
Layer 12: ReLU
Layer 13: Max pooling layer with a (# of time splits for a data point)/8 x 1 pool size. (This is HxW)
Layer 14: Dropout layer with a dropout probability of 0.2
Layer 15: Fully connected layer with 2 nodes.
Layer 16: Softmax

Note: “same” padding is edge padding added to an input image so that the output image size will be equal to the input image size