Real-Time, AI-Assisted Flame Stability Analysis for Flame Spray Pyrolysis

Jessica Pan[1], Noah Paulson[2], and Marius Stan[2]

[1] Princeton University, Princeton, New Jersey, United States
[2] Argonne National Laboratory, Lemont, Illinois, United States

Abstract. Flame spray pyrolysis (FSP) is an experimental process used to synthesize nanoparticles through the ignition of a solvent as it is ejected from a burner. However, current limitations in the understanding of how to consistently achieve a stable flame is impeding the reliable production of particles using this method. Therefore, streamlining the flame spray process using machine learning and artificial intelligence on FSP video feed to detect unstable flame conditions will contribute to fewer wasted resources as well as a safer working environment. We first determine a way to quantitatively decide whether or not the burner flame is stabilized by analyzing the anchor point of the flame. This method can be used to label data for both unsupervised and supervised learning techniques or applications, such as principal component analysis or image detection. In doing so, we can track and classify FSP flame conditions in real time, and alert users if an unstabilized flame state is achieved at any point in the synthesis process. This research has the potential to more efficiently manage manufacturing processes using machine learning and computer vision.

Keywords: Flame Spray Pyrolysis · Artificial Intelligence · Computer Vision · Machine Learning · Image Classification

1 Introduction

Flame spray pyrolysis (FSP) is a cost-effective, versatile and scalable synthesis process used generate make powder-like nanoparticles by combusting a solvent loaded with precursors [1]. In our setup, the precursor spray is expelled from a nozzle and ignited by pilot flames. These nanoparticles are oxides, salts, or metal/alloy nanoparticles that are synthesized at a low cost using this method for manufacturing purposes [2].

Because FSP is an novel and experimental process, we do not know how the nanoparticles nucleate and grow, nor can we create nanoparticles of the desired size consistently [3]. In other words, the parameters required to produce a target yield of nanoparticles are still unclear because we do not know the mechanism by whence they form [3]. However, flame spray pyrolysis will not produce a favorable yield of nanoparticles if the burner flame becomes unstabilized [4]. This is because the completion of combustion of an unstable flame would become uncertain, and thus solvent may leave the process without having been fully combusted. Not only does this waste valuable time and resources, but this can compromise the safety of the setup due to residue buildup of unignited solvent. Detecting and lessening the frequency of unstable flame conditions in flame spray pyrolysis creates a more efficient and safer working environment.

In the past, researchers like Dasgupta et al. [5] have used methods such as computational fluid dynamics modeling to better understand the mechanism behind nanoparticle synthesis in flame spray pyrolysis. This physics-driven approach have provided us with a keen understanding of the forces at play during the synthesis process by investigating the fluid dynamics of the spray, air, and flame around the FSP burner [5]. While Dasgupta and her team's research focuses modelling the dynamics of the FSP flame, in practice such a method would be too inefficient to provide scientists with a real-time evaluation of their flame condition. We aim to devise a method of alerting scientists of unstable (and therefore also unsafe and wasteful) flame conditions as soon as they happen. This would save the need for human monitoring (and so researchers will be able to better use their time elsewhere) and improve laboratory/factory safety.

Thus, the specific purpose and application of this project is to optimize burner flame stability of flame spray pyrolysis in near-real-time (within seconds) by processing video feed of the burner. This could allow for a program to autonomously learn from the process and correctly label the flame state faster and with less error when compared to its human counterparts. We do this in two ways: by using principal component analysis (unsupervised machine learning) and a convolutional neural network to create an object detection classifier (supervised machine learning).

2 Methodology

2.1 Data

Our data consisted of 53 full-color, 10-second video clips of flame spray pyrolysis burning a simple ethanol (EtOH) solvent with no solute in a controlled environment. This is imperative to the purpose of our research, which will focus on detecting features from FSP video clips to predict the stability of the flame in the videos. The videos were taken from a webcam stationed above the FSP setup so as to capture footage of the burner cap, pilot flames, and main burner flame when it is operating. The screen of the desktop receiving live webcam feed was then recorded to produce the video clips used for our research.

Additionally, 11 human experts in the field of combustion (Argonne researchers, Princeton professors and graduate/undergraduate students) were each asked to view the 53 FSP video clips and classify the state of the flame (stable, unstable, or uncertain). They did not collaborate or discuss their evaluations with others. These responses will be considered the standard against which we compare our results.

2.2 Tools

Python was used for machine learning and image classification, with extensive use of its Scikit-learn, SciPy, and OpenCV libraries. TensorFlow and CUDA were also used to train the RCNN we used to detect and classify images in video feed. The object detection classifier code was based off of the tutorial on the EdjeElectronics GitHub [6].

3 Stability Classification

A computational approach to flame stability classification is necessary for this project to generate evaluations that require minimal human (manual) interference. The area closest to the nozzle where the flame originates is called the anchor point. Momentary extinction events occurring at the flame's anchor point (where the flame front starts propagating downstream) is an indication of flame instability. In other words, on the video feed such extinction events would appear as if the flame is flickering out and separating from the nozzle (see Fig. 1's comparison of a flame that is experiencing extinction near its origin with a flame that is stabilized).

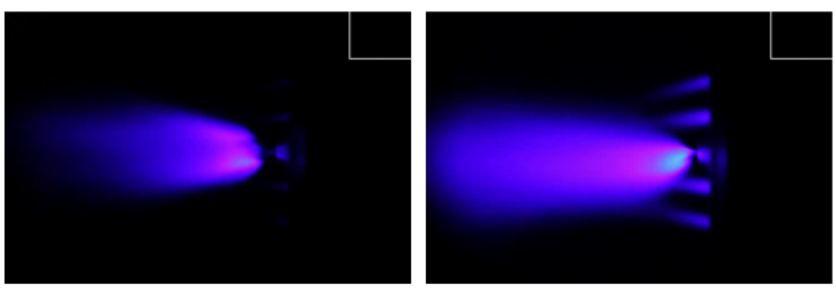


Fig. 1: Unstabilized flame (left) and stabilized flame (right). The unstabilized flame front is no longer attached to its origin (the nozzle towards the right of the image), while the stabilized flame front is.

Our approach for quantitatively determining flame stabilization (we will call it the bounding box luminosity approach) involves analyzing the luminosity (a measure of pixel brightness) of pixels within a bounding box region of the burner flame (see Fig. 2). Luminosity is digitally stored as an integer value between 0 and 255, and the bounding box is (50px tall, 30px wide) around the flame's anchor point. We know that the area closest to the nozzle exhibits great fluctuations in luminosity when the flame is not stabilized due to the occurrence of momentary extinction events, where the flame is no longer anchored to the nozzle. We have written a program to detect such fluctuations in luminosity and used it to classify the flame stability of our video clips.

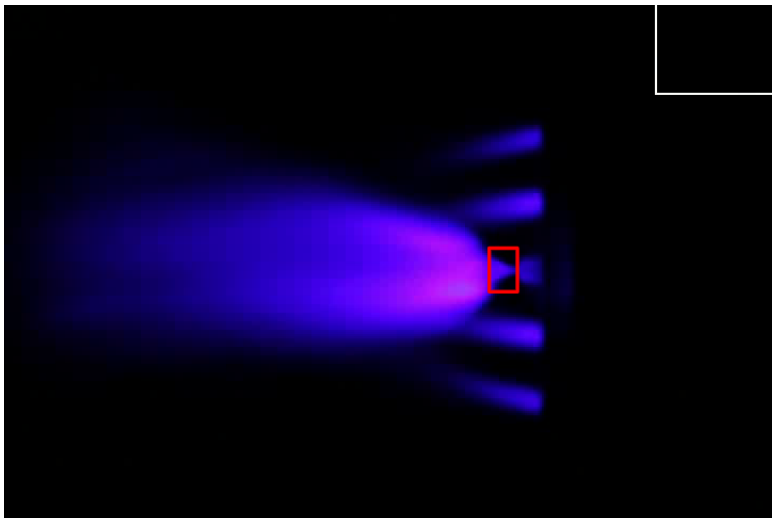


Fig. 2: Bounding box around flame anchor point. The pixels inside the red rectangle are the pixels whose luminosity will be measured and used to classify the stability of the flame.

We averaged the luminosity of all the pixels in that bounding box across all of the frames in a given video clip. Each frame's deviance in luminosity from the average luminosity of all of the frames in the video is then plotted. If the fluctuation in luminosity deviated more than 25% from the average, the flame in the clip is considered unstabilized. If it deviates more than 15% from the average, we consider the flame neither stable nor unstable and give it an uncertain rating. Otherwise, the flame is considered stabilized. Examples of this process as applied video clips are shown in Fig. 3 and Fig. 4. The threshold percentages were chosen arbitrarily from our own visual inspection and classification of the video clips in our dataset.

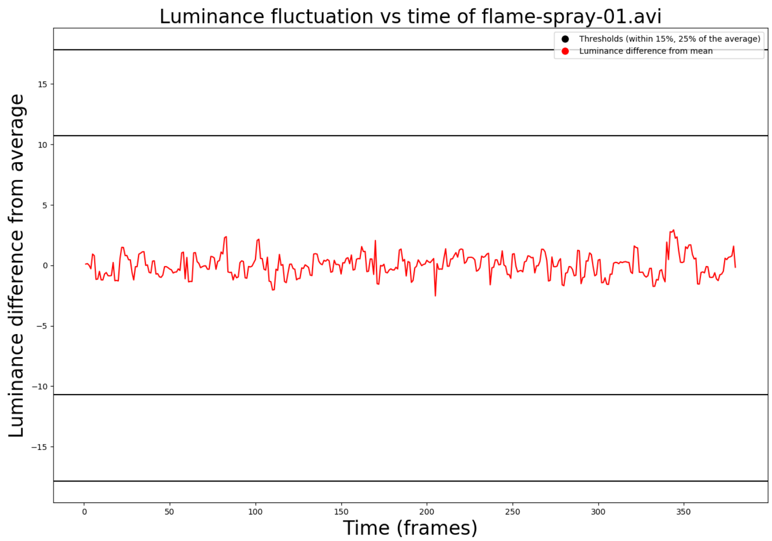


Fig. 3: Bounding box luminosity fluctuation plot for what we classify as a stabilized flame (for video clip flame-spray-01.avi).

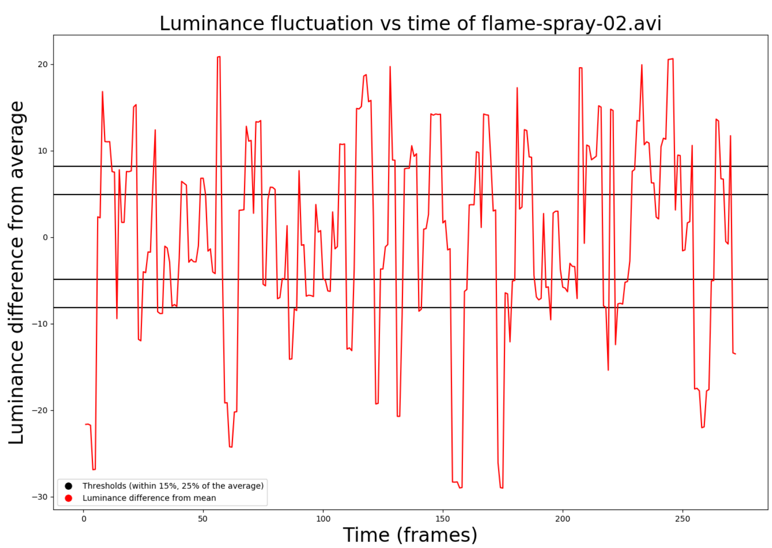


Fig. 4: Bounding box luminosity fluctuation plot for what we classify as a stabilized flame (for video clip flame-spray-02.avi).

3.1 Unsupervised Machine Learning

After labelling the flame spray pyrolysis videos, using the classification method detailed in Stability Classification, we use an unsupervised machine learning approach to determine whether or not an algorithm is capable of finding a combination of different features which correspond to different classes of flame stability. Our features (described in more detail later) are composed of bounding box pixel luminosities in consecutive frames of video feed (to better capture the dynamics of the flame).

3.2 Principal Component Analysis

Principal component analysis is a common dimensionality reduction technique that compresses data from an $n \cdot p$-dimensional space into a much smaller one while minimizing information loss [7]. It extracts features from the data and combines them in such a way (using covariance matrices, eigenvectors, and eigenvalues) such that we only keep the most important features of all of the variables [8]. The new variables (combinations of the extracted features) are called principal components, and they are independent of each other, which means the original data points may undergo the same transformation and be plotted with the principal components as the axes.

We will be using principal component analysis in our unsupervised machine learning approach to determine if there is some feature or combination of features within our data that may explain flame stability. If there exists such a combination of features, then it is much more easily visualized and understood in a space with fewer dimensions.

3.3 Application of Principal Component Analysis

Our original feature space is 30x1500-dimensional and consists of a vector containing the luminosity of every pixel in the bounding box region of 30 consecutive frames/images of video. Using principal component analysis, we are able to reduce down to merely 2 dimensions (principal components) while still capturing about 70% of the variance of the original data (Fig. 5).

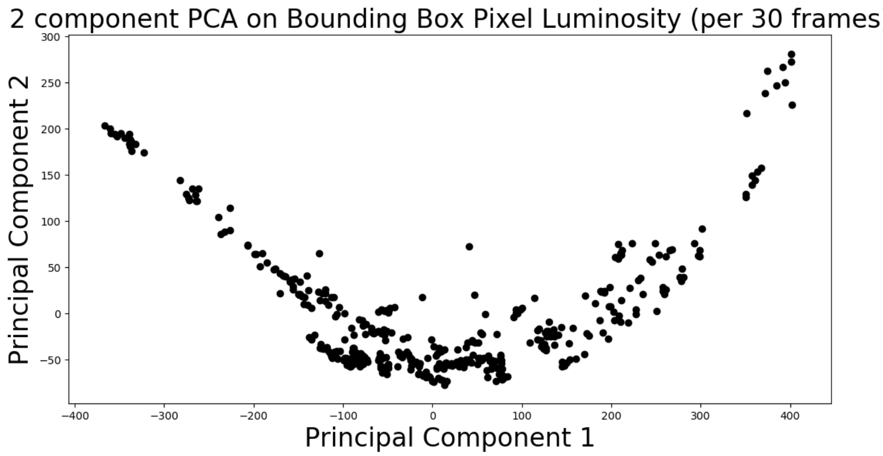


Fig. 5: PCA results for bounding box pixel luminosity. Each point represents 30 consecutive frames in a given video clip, or one second of footage.

By coloring the frames according to their assigned classifications using the method from Stability Classification, we can see that there is a cluster of points that are nearly all deemed unstable by our prior stability classification. In other words, the closer a point lands to the red 'unstable' cluster seen in the graph, the higher the chance it represented a unstabilized-flame video clip (Fig.6).

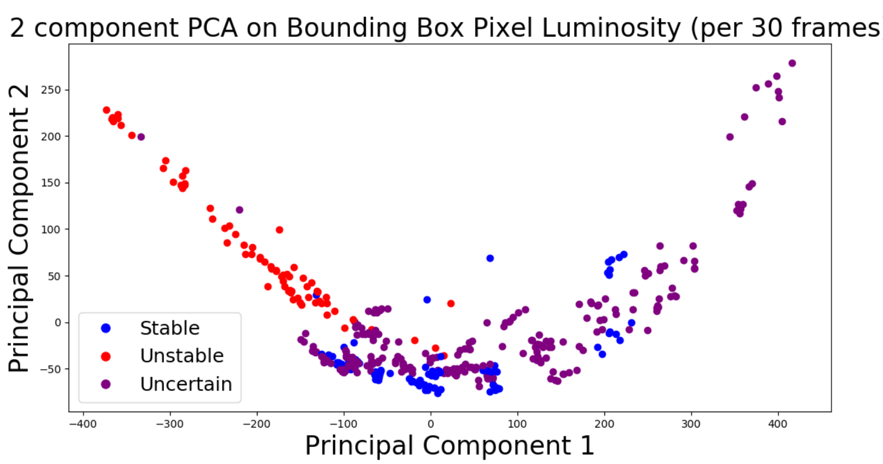


Fig. 6: PCA results for bounding box pixel luminosity. Each point represents 30 consecutive frames in a given video clip, or one second of footage. Points are colored by their classification: red points represent video clips of flames that are not stable, blue represents stable flames, and purple represents an 'uncertain' classification.

Even when entirely new 30-frame video snippets are introduced as data points on the PCA graph, it is possible to classify their stability by determining where they lie in relation to our known stable/unstable/uncertain regions. We apply k-means clustering (with three clusters, one each for 'uncertain', 'unstable', and 'stable') to our plot in Fig.5, and plot the centroids of those regions in black.

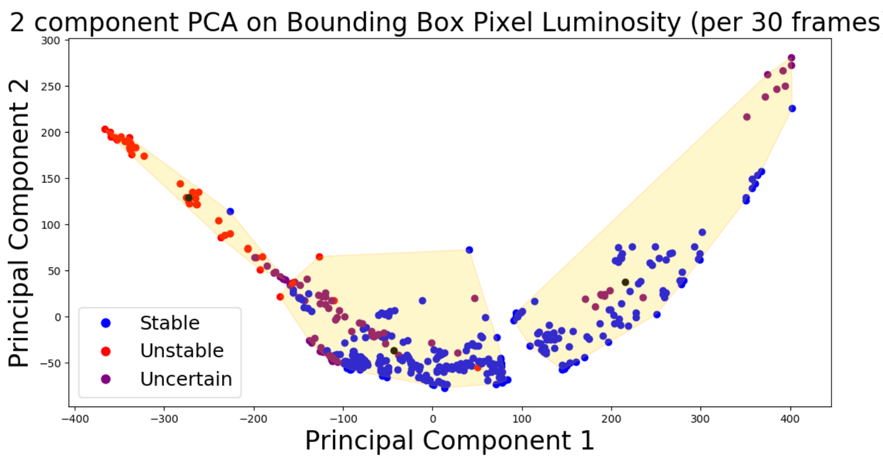


Fig. 7: PCA results for bounding box pixel luminosity. Each point represents 30 consecutive frames in a video clip, which corresponds to one second of real-time activity. Three clusters (centroids plotted in black and a convex hull forms their outline) exist within our data; the one of note is the cluster containing most of the video data labelled 'unstable' to the left.

As seen in both Figure 8 and Figure 9, we introduce two new samples and plot them on the two existing principal component axis in green. If a point lies closer to the centroid of the 'unstable' region than the other two centroids, then the new clip will also be labeled 'unstable'. Otherwise, it will be labeled 'stable'. However, in practice we are not plotting anything during this operation. Instead, we would only need to feed a 30-second clip into the algorithm which will then transform it to our principal-components space. It will then then use k-means to identify which cluster the new data belongs to.

Fig. 8: PCA results for bounding box pixel luminosity. Each point represents 30 consecutive frames in a video clip, which corresponds to one second of real-time activity. Black points are the centroids of the three k-means regions. The green point is a new sample that has been projected on top of the pre-existing graph. This one appears closer to the centroid in the 'stable' region, and thus the video clip will be labeled 'stable'.

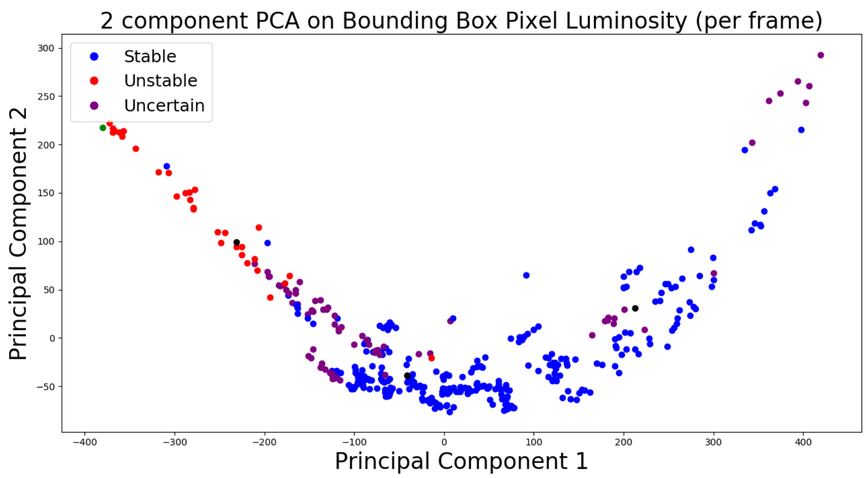
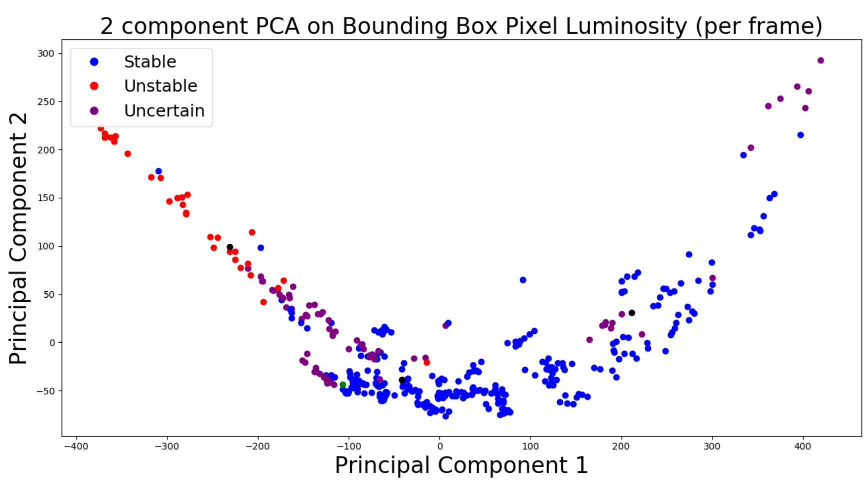


Fig. 9: PCA results for bounding box pixel luminosity. Each point represents 30 consecutive frames in a video clip, which corresponds to one second of real-time activity. Black points are the centroids of the three k-means regions. The green point is a new sample that has been projected on top of the pre-existing graph. This one appears closer to the centroid in the 'unstable' region, and thus the video clip will be labeled 'unstable'.



In summary, to achieve a near-real-time analysis of the flame's stability using this method, footage (in the form of vectors of pixels) must be passed into this algorithm and plotted on the existing PCA graph. Its position will be compared relative to the three centroids on the graph ('uncertain', 'unstable', and 'stable') and the clip will adopt the classification of the centroid it was closest to. An update regarding the stability condition of the FSP flame can be provided every second.

3.4 Supervised Machine Learning

Meanwhile, object detection and image classification are prevalent and necessary techniques used in artificial intelligence and computer vision. It can be used to nearly immediately identify features or objects given either a still image or in video feed. Thus, we conjecture that it also can be an effective way of classifying stabilized and unstabilized flame conditions in a video.

To show this, we use TensorFlow's RCNN Inception V2 model [9] to detect and label different features in our videos. The algorithm receives an input image then proceeds to extract several regions from it, using those regions as inputs for a convolutional neural network [10]. The neural network then extracts features from those regions and tries to classify them into categories that are dictated by a user. We specify the "burner cap", "pilot flame", "unstable [flame]", and "stable [flame]" categories. The classification process itself is done through training the model's support vector machine. To train the model, we incrementally sample all of the video frames from our dataset (saving 1 frame in every 30) and label all of the relevant features spotted in the video (pilot flame, unstable or stable flame state, and burner cap) we found in each of the chosen frames. This resulted in a set of approximately 570 human-labelled images, twenty percent of which are allotted to a test set (used to assess the performance of our model) and the rest of which are given to the training set (which is used to modify the weights used in the algorithm to predict the eventual classification). The model trained for about thirteen hours, after which it was given unseen footage to classify (such as the image represented in Fig. 10)

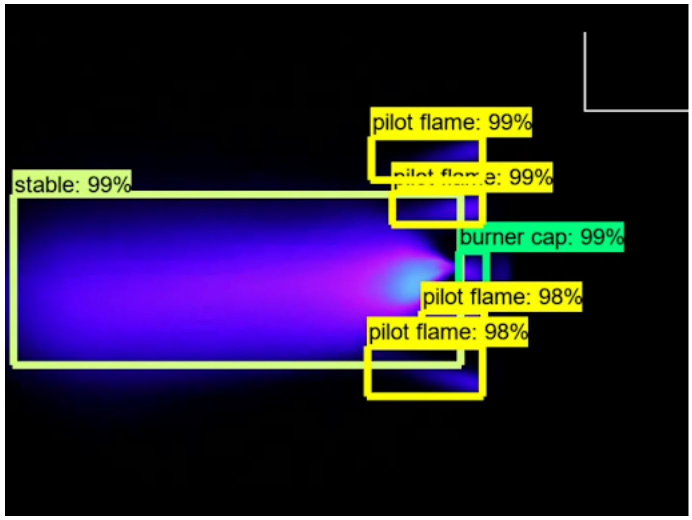


Fig. 10: Image with features classified by trained object detection model.

The model was able to recognize instances of both stable and unstable flame states; the numerical labels in each bounding box represent the percentage confidence that the features detected are indeed accurately labeled as their respective elements. Additionally, we modified the code so that the warning of "Unstable flame state detected!" appears in the terminal every time the classifier detects an instance of an unstabilized flame. In the future, this could become an auditory alert on the interface in charge of running flame spray pyrolysis. This serves as example of how even this simple model could save humans time and work by calling the attention of scientists only when a fault is detected and needs to be fixed.

4 Results

To measure our classifications' success, we compare the predictions generated by our two approaches with human classifications. Eleven scientists, professors, and students who are familiar with and have worked directly with combustion took part in looking through our dataset and they labelled each video one of three categories: 0 for unstabilized, 1 for uncertain, and 2 for stabilized flame state. The responses for each video clip is averaged between the eleven subjects. In order to directly compare the human classifications with the machine-generated predictions, we consider any averaged classification greater than 1 (more than 'uncertain' stability classification) as 'stabilized' (2), and any averaged classification less than 1 as 'unstabilized' (0) (if the average was exactly 1, we gave the human classification an 'uncertain' value of 1). The results of the human classification and our predictions are plotted in Figure 9.

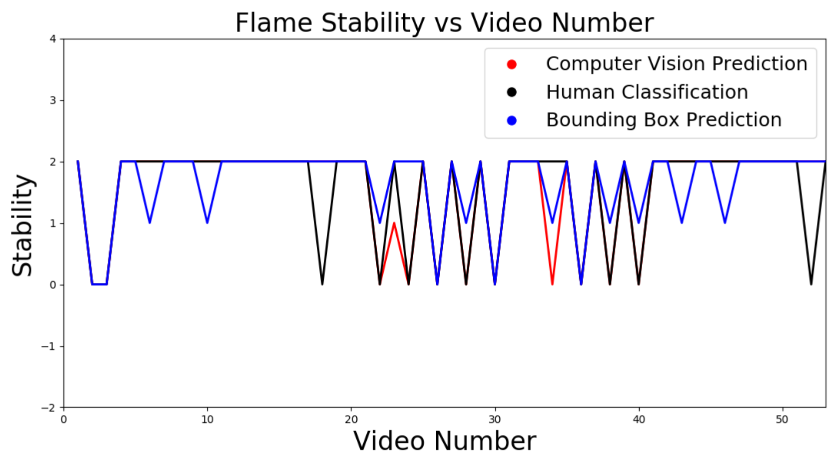


Fig. 11: Average human classification of flame stability for each video in the dataset vs the predictions generated by the bounding box and computer vision approaches in this study. Stability values on the y-axis correspond to:

For 45 of the 53 videos, or in about 85% of the cases, the prediction achieved from our bounding box prediction (used in unsupervised machine learning) matches that of the human classification. For the computer vision approach, the accuracy is even higher, with human experts and and model agreeing in 92% of the cases.

5 Discussion

5.1 Conclusion

The overall goal of the Near Real Time Optimization for a Manufacturing Process project, led by Argonne National Laboratory's Dr. Marius Stan, is to create a cohesive team of humans and software to perform real-time optimization of the flame spray pyrolysis synthesis process. In their approach, data from experiments feeds a machine learning model, which in turn generates predictions for future trials. This creates a cycle of repeated learning and observations. Our approach finds a solution through computer vision and machine learning via easily accessible (webcam) video feed of the FSP burner flame.

Our project's specific objective is to minimize the amount of resources and time wasted by generating sub-optimal yields of target particles during flame spray pyrolysis. Real-time optimization of manufacturing methods can boost productivity, enhance workflow, improve safety, and even eventually perform better than human work, so research in this area has the potential to greatly improve the production of resources needed for development in areas such as energy.

Using a bounding box indicator on images to detect luminosity fluctuations near the nozzle of a burner flame makes it possible to quantitatively determine flame stability and use it to classify flame spray pyrolysis footage and images. By applying principal component analysis for two principal components, we are able to isolate clusters of events determined to be in relative groups of stability; giving the algorithm additional points (in the forms of individual frames) and assigning them to the nearest cluster would provide a near-real-time means of generating stability classifications for new data. Additionally, it is possible to train an object detection classifier for the purpose of recognizing and labeling features in videos of flame spray pyrolysis, and a warning can be sent to a user if the flame should reach an unstable state.

However, both methods have their limitations and drawbacks. Principal component analysis is not generalizable to different types of video feed. If the angle, shape, or other aspect of the video itself is changed, then humans are required to re-define another bounding box around the nozzle manually in order to run the analysis again. This is because the bounding box luminosity stability classification is dependent on the luminosity of pixels near the burner nozzle, which is not automatically defined by the program and requires human intervention.

Generalizability is a lesser concern for our image detection and classification approach – the model should be able to classify flame stability despite some changes to the way the video is taken. This is because detects patterns (features) in each frame to classify images rather than focusing on a specific part of the image itself. Currently, our model has only been trained on video clips taken from the same direction, of the same proportions, etc. As a result, the accuracy of our model's predictions will likely decrease on videos of different proportions/directions/angles without training on a more diverse set of data.

Two factors may also contribute to lowering the accuracy of the predictions generated by our programs as compared to expert evaluations. Cases where humans consider video feed of a flame stable while our model does not may be attributed to the human subjects' inability to detect an extinction event that lasts too short a time. Video clips are analyzed frame-by-frame (30 frames per second) by the program, and even if extinction on a very small time scale occurs, the program will be able to pick up on it. Additionally, the thresholds for luminosity fluctuation set in our Stability Classification section were also set arbitrarily using the researcher's evaluation of stability in video clips. Changing them may improve our predictions as well.

5.2 Future Work

Further progress can certainly be made to this research. The current model used for object detection is slow and there is a subtle yet noticeable lag between an event and its classification in the program. A different object detector such as the YOLO (You Only Look Once) model could provide a much faster means of identifying features within the video [11]. Moreover, rather than just alert researchers of unstable flame conditions, achieving autonomous correction of the parameters of the flame to revert back to a stable flame state is considered a reach goal of this project. This would mean that a researcher could leave the flame unattended for longer periods of time, as a program can simply stabilize the burner flame should the need arise without any human intervention.

6 Acknowledgements

Thank you to Marius Stan and Noah Paulson for advising and contributing to all parts of this project and offering guidance, support, and constructive insight. Thank you, Debolina Dasgupta and Joe Libera, for acquainting me with flame spray pyrolysis, helping me collect my data, and offering advice on how I can approach the problem. A note of gratitude to Professor Jia Deng as well for being my independent work advisor and guiding me to the resources and ideas I needed to complete this research.

References

[1] Madler, L. et al. “Controlled Synthesis of Nanostructured Particles by Flame Spray Pyrolysis.”Journal of AerosolScience, 2002, doi:10.1016/s0021-8502(01)00159-8.

[2] Nunes, D. et al. “Synthesis, Design, and Morphology of Metal Oxide Nanostructures.” Metal Oxide Nanostructures, 2019, pp. 21–57., doi:10.1016/b978-0-12-811512-1.00002-3.

[3] Hinklin, T., et al. “Liquid-Feed Flame Spray Pyrolysis of Metalloorganic and Inorganic Alumina Sources in the Production of Nanoalumina Powders.” Chemistry of Materials, vol. 16, no. 1, 2004, pp. 21–30., doi:10.1021/cm021782t.

[4] Solero, G. et al. ”Synthesis of Nanoparticles through Flame Spray Pyrolysis: Experimental Apparatus and Preliminary Results.” Nanoscience and Nanotechnology, 2017, pp. 21-25, doi: 10.5923j.nn.20170701.05.

[5] Dasgupta, D. et al. “Computational Fluid Dynamic Modeling of the Flame Spray Pyrolysis Process for Silica Nanopowder Synthesis.” Journal of Nanoparticle Research, vol. 17, no. 7, 2015, doi:10.1007/s11051-015-3109-z.

[6] Juras, Evan. “TensorFlow Object Detection API Tutorial Train Multiple Objects Windows 10”EdjeElectronics,https://github/com/EdjeElectronics/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10.

[7] Jolliffe, Ian T., and Jorge Cadima. "Principal Component Analysis: a Review and Recent Developments." Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 374, no. 2065, 2016, p. 20150202., doi:10.1098/rsta.2015.0202.

[8] Brems, Matt. "A One-Stop Shop for Principal Component Analysis." Medium, Towards Data Science, 10 June 2019, towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c.

[9] Tensorflow. "Tensorflow/Models." GitHub, github.com/tensorflow/models/blob/master/research/object$_{d}etection/g3doc/detection_{m}odel_{z}oo.md$.

[10] Xu, Joyce. "Deep Learning for Object Detection: A Comprehensive Review." Medium, Towards Data Science, 29 Sept. 2017, towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9.

[11] Gandhi, Rohith. "R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms." Medium, Towards Data Science, 9 July 2018, towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e.