# arcjetCV: automating arcjet video analysis

Magnus A. Haw[1], Alexandre M. Quintart[2]

[1]NASA ARC-TSM, Moffett Field, CA
[2]Flying Squirrel, Belgium

Presented By
Magnus A. Haw

Thermal & Fluids Analysis Workshop
TFAWS 2023
August 21-25, 2023
NASA Goddard Space Flight Center
College Park, MD

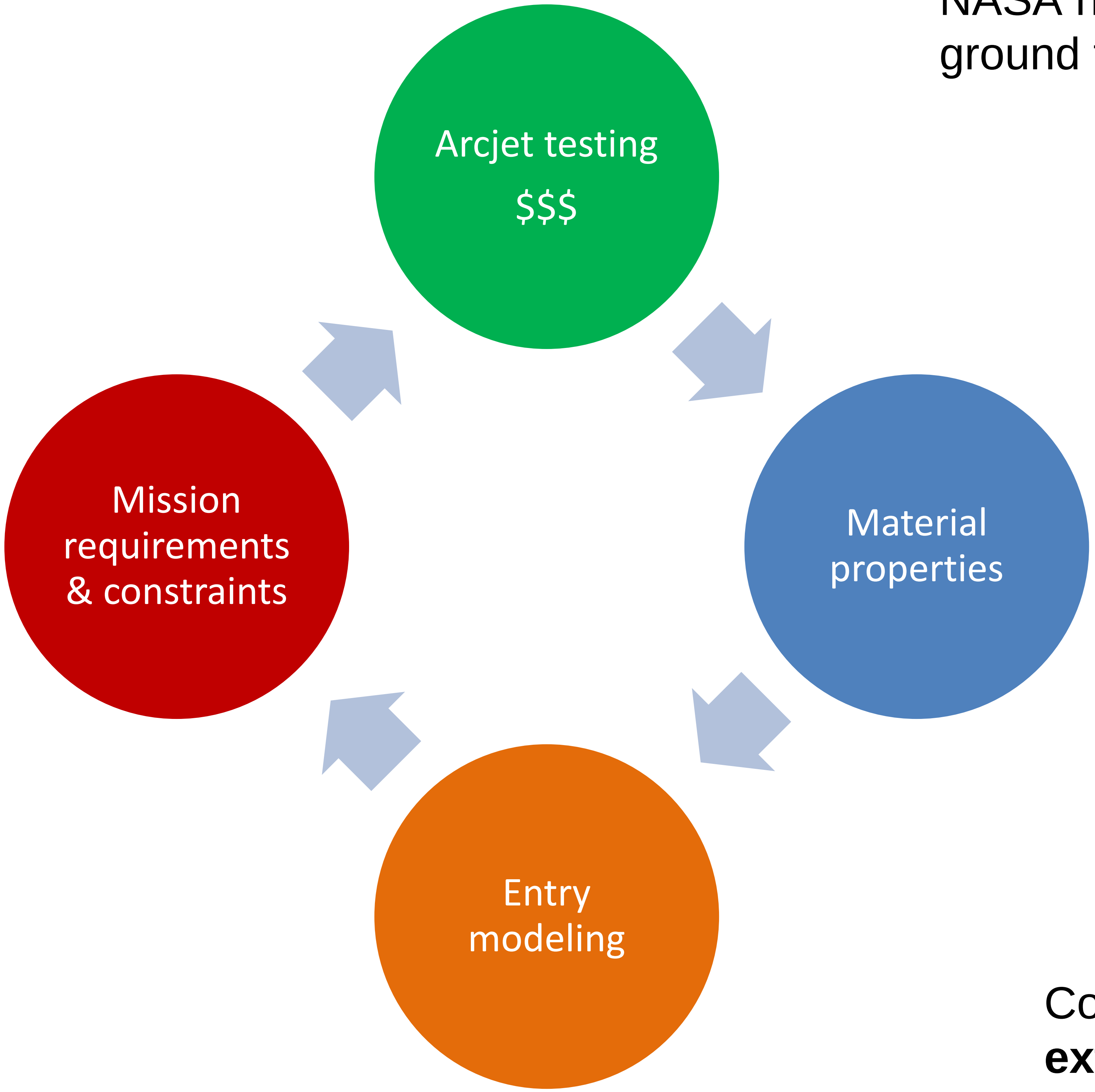
TFAWS23-AE-10
Aeroscience session #2

Thursday, Aug 24th,
2:30 PM - 3:00 PM
Room: Salon D

# Outline

- Introduction
- Software
- ML models
- Discoveries

# Introduction



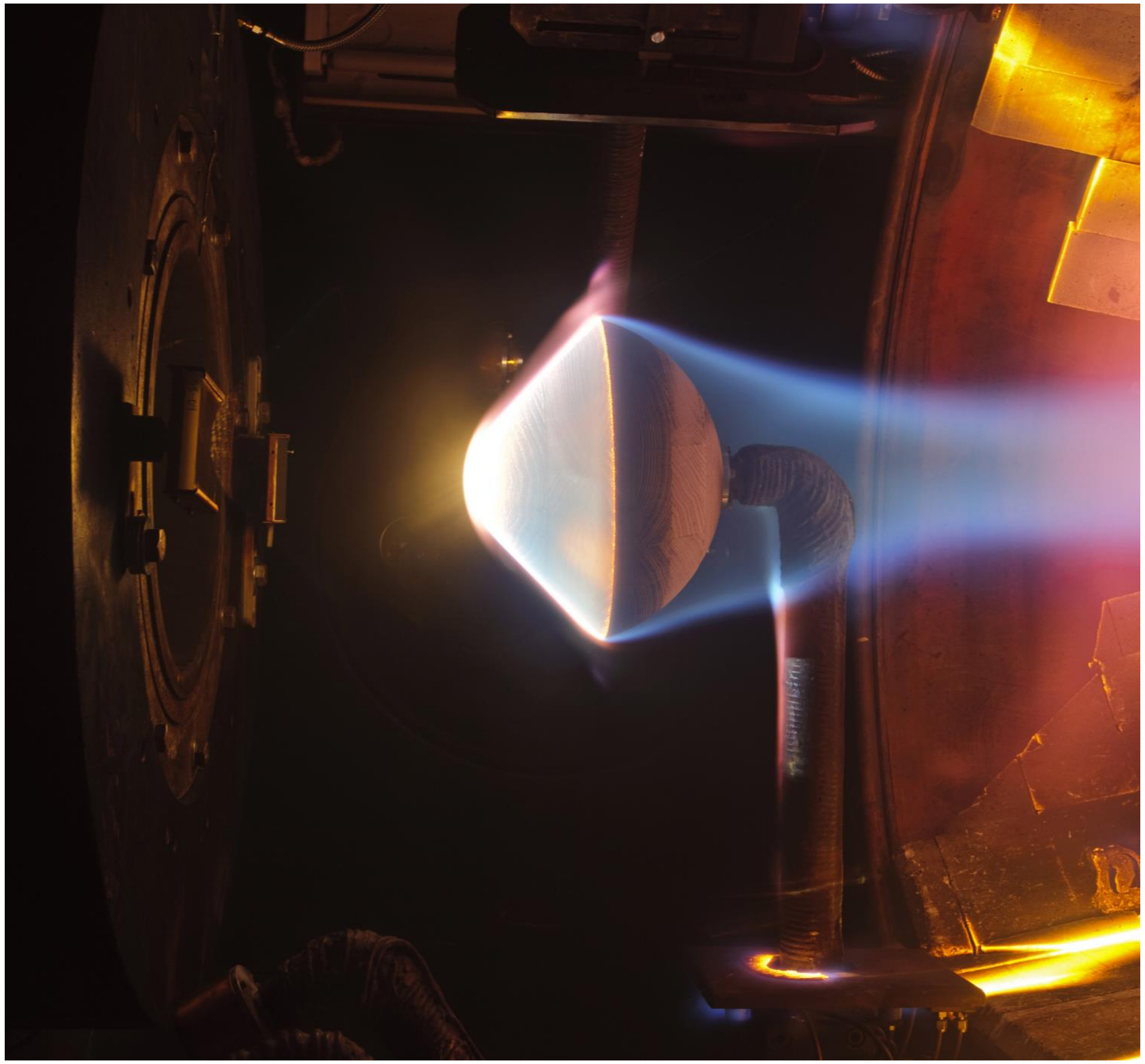

NASA heatshields require extensive ground testing and modeling

Ground tests are expensive ~$100k/sample and complex but necessary to evaluate material performance

Consequently, **every arcjet test should extract as much material data as possible**

# Arcjet testing

Def. arcjet: plasma wind tunnel



arcjet test chamber showing sample under test

1. Material samples (2-6” ⦰) are fabricated and instrumented with internal thermocouples

2. Samples are placed in test chamber and exposed to high enthalpy flow from arcjet

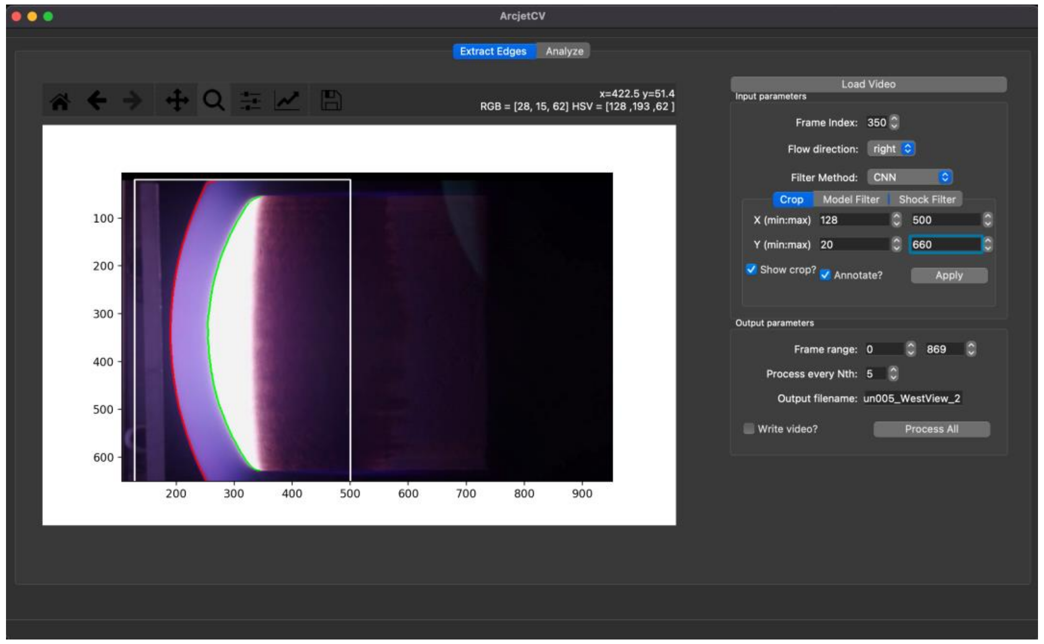
3. Material properties are inferred from combination of calorimetry, CFD, thermocouple data, pyrometers, and surface recession

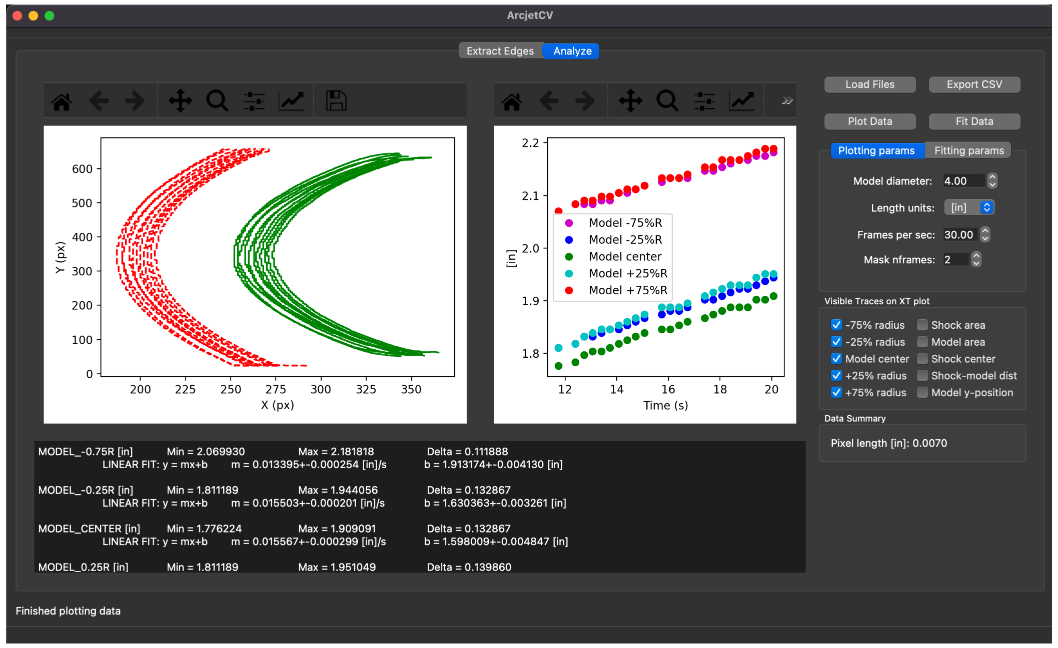
# Problems

A. Material samples are only 3D scanned before and after a test, no in-situ sensing
B. Tracking recession from video is sufficiently difficult that it does not occur often
C. Non-linear recession and complex behaviors (e.g., melt, swelling, shrinkage) cannot be quantified without time-resolved recession

# Solution: arcjetCV

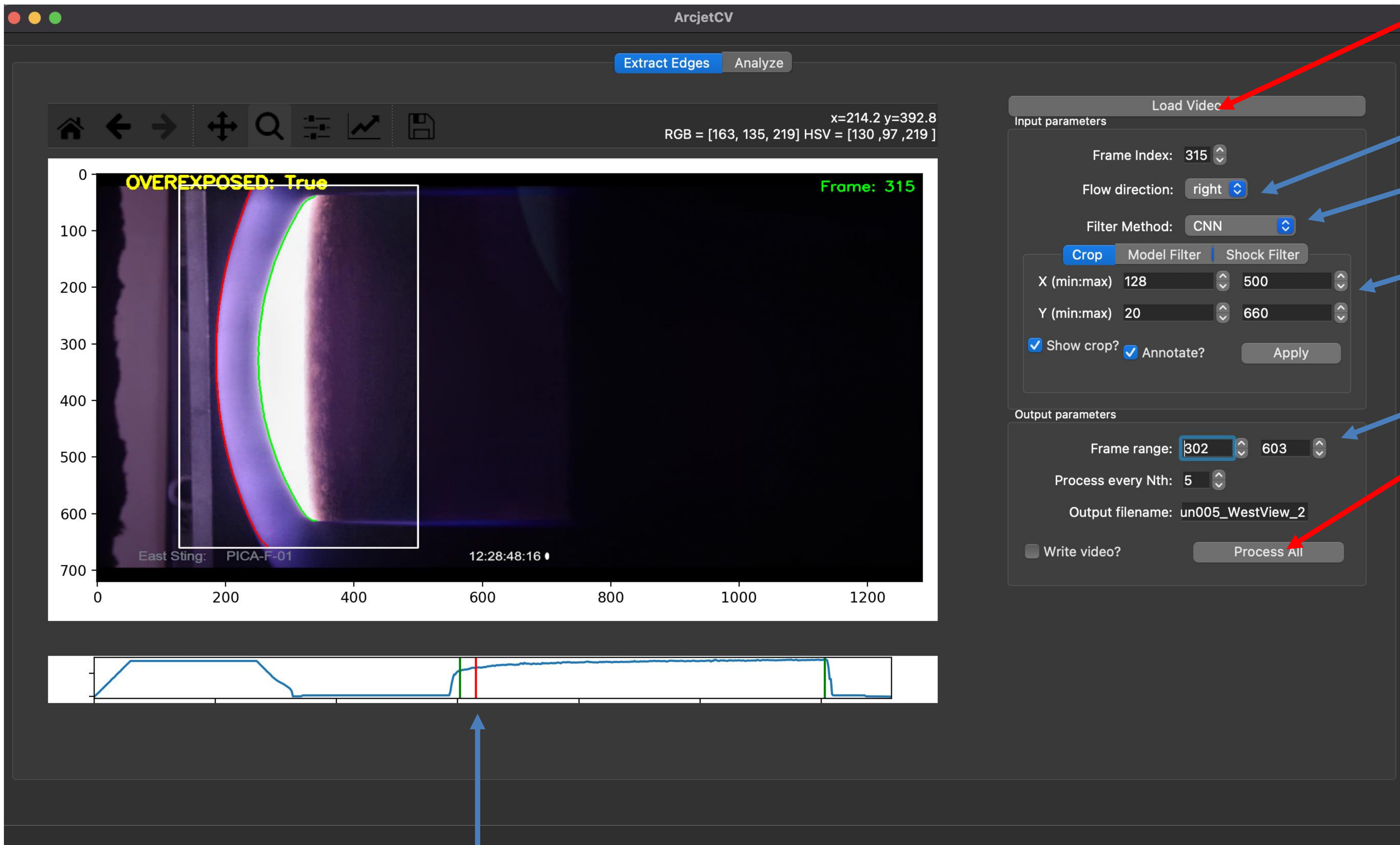
A. Computer vision & machine learning automate data pipeline
B. Graphical user interface enables anyone to process video
**C. <u>Result</u>**: new time-resolved recession tracking for all samples



# Software: 100% Python

# Desktop Interface




1. Load a video

- Select flow direction
- Select filter method
- Crop & Filter parameters
- Output settings

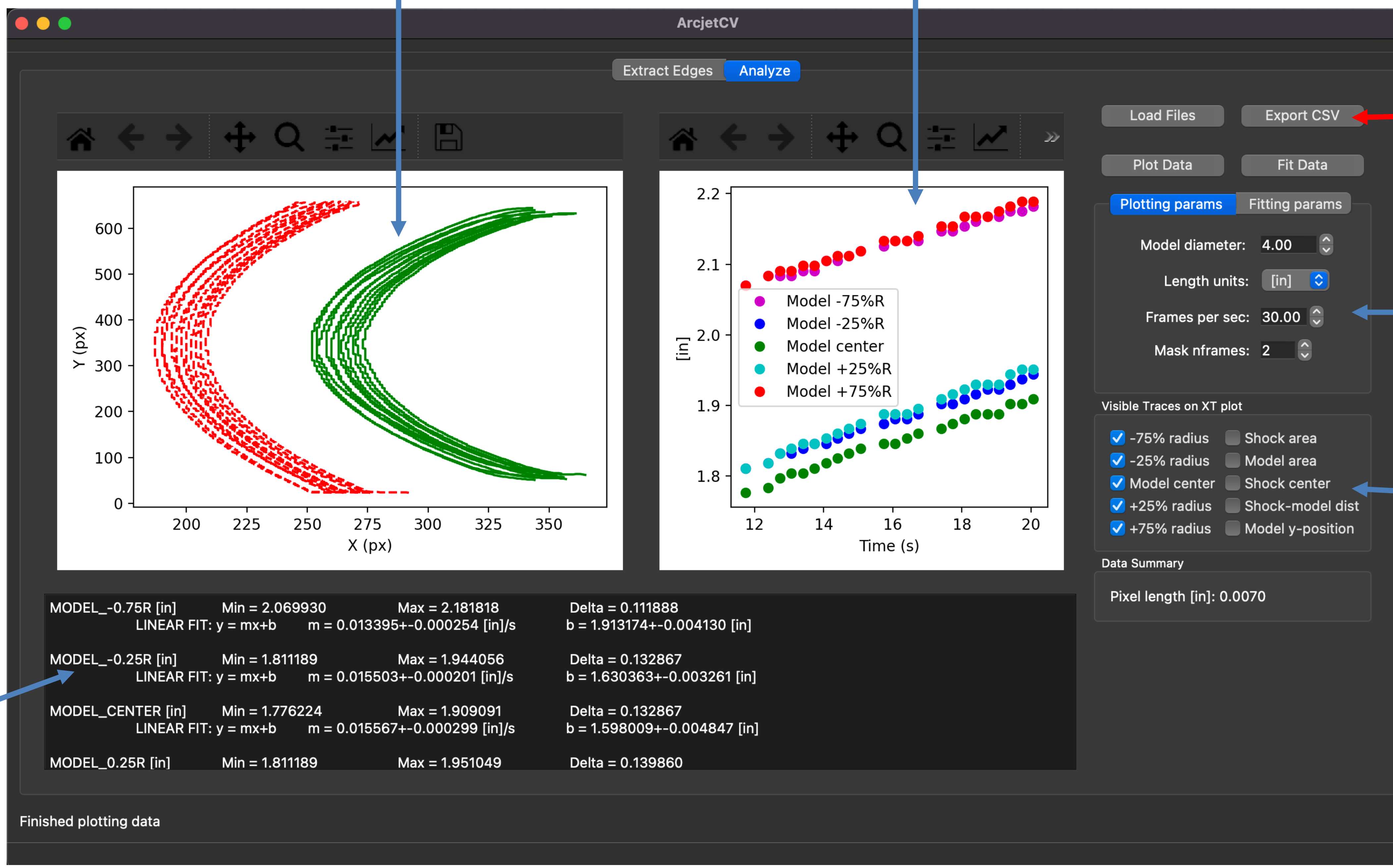
2. Process all frames

Clickable navigation bar: displays integrated frame intensity, start/stop frames (green lines) and current frame (red line)

# Desktop Interface

XY traces of shock and the sample edge

Time dependence plot




3. Export

Parameters

Plot options

Linear Fits

# Processing Pipeline

## Preprocessing

Estimate first/last frames
Estimate ROI
Guess flow direction
Histogram equalization
RGB-> HSV conversion
Crop frames to ROI
Create/save metadata log file

## Frame segmentation

Segment image into 3 classes
Extract leading edges (sample/shock)
Save edge points to JSON output
Extract top/bottom positions
Extract centerline position
Extract radial positions (+- 25%, 75%)
Extract shock standoff at centerline
Extract sample/shock areas
Save extracted series to JSON output

## Post-processing

Filter out outliers
Convert pixels to units
Plot time series data
Plot XY edge traces
Fit time series data

These steps were improved by using machine learning models

# ML Models

- VGG16-UNet (CNN model for high accuracy)
- Decision tree model (for fast segmentation)
- Time segmentation (1D CNN for detection of start/stop frames)

How arcjetCV automates the annoying/time consuming stuff

# Convolutional neural net (CNN)

CNNs are neural nets designed to recognize patterns in images

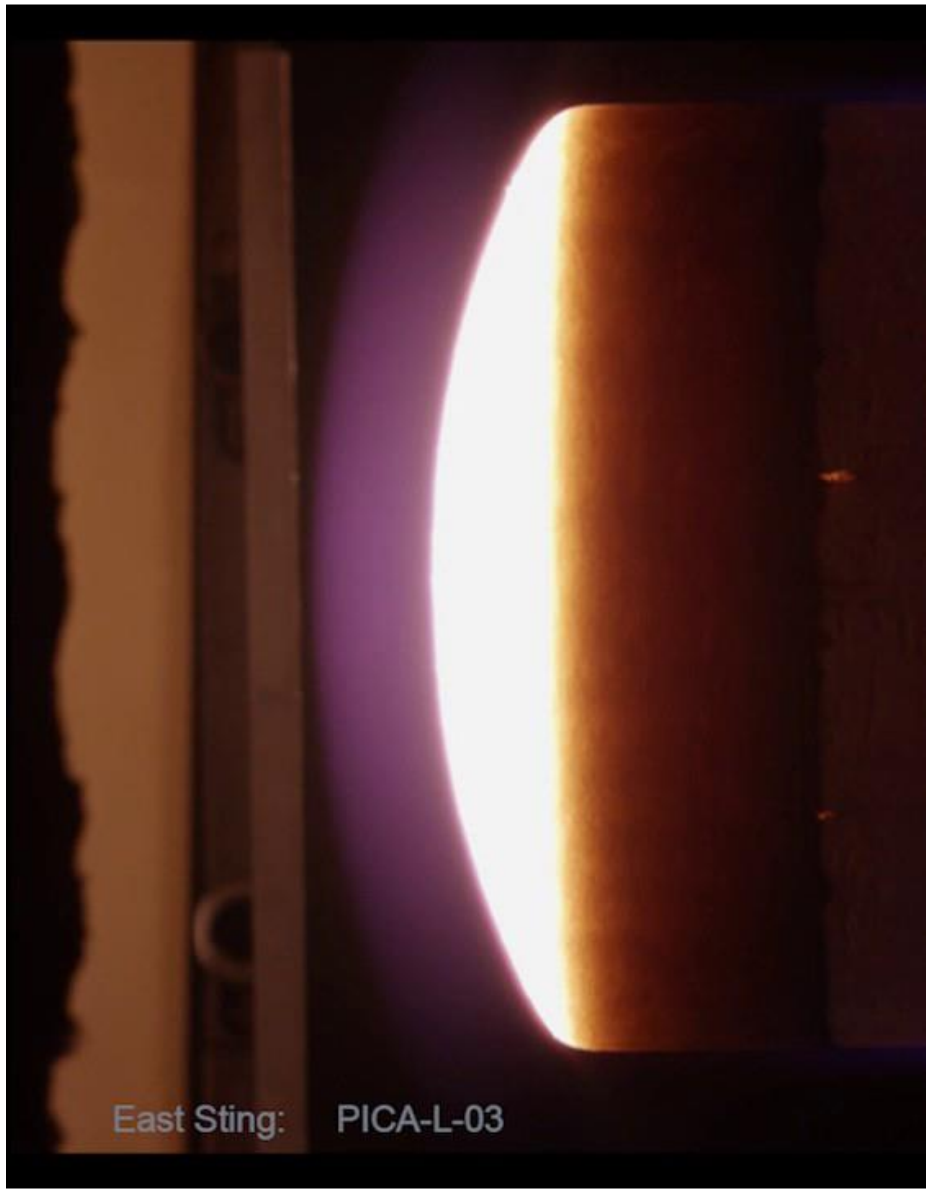
- typically trained on manually labeled images
- robust performance compared to other methods

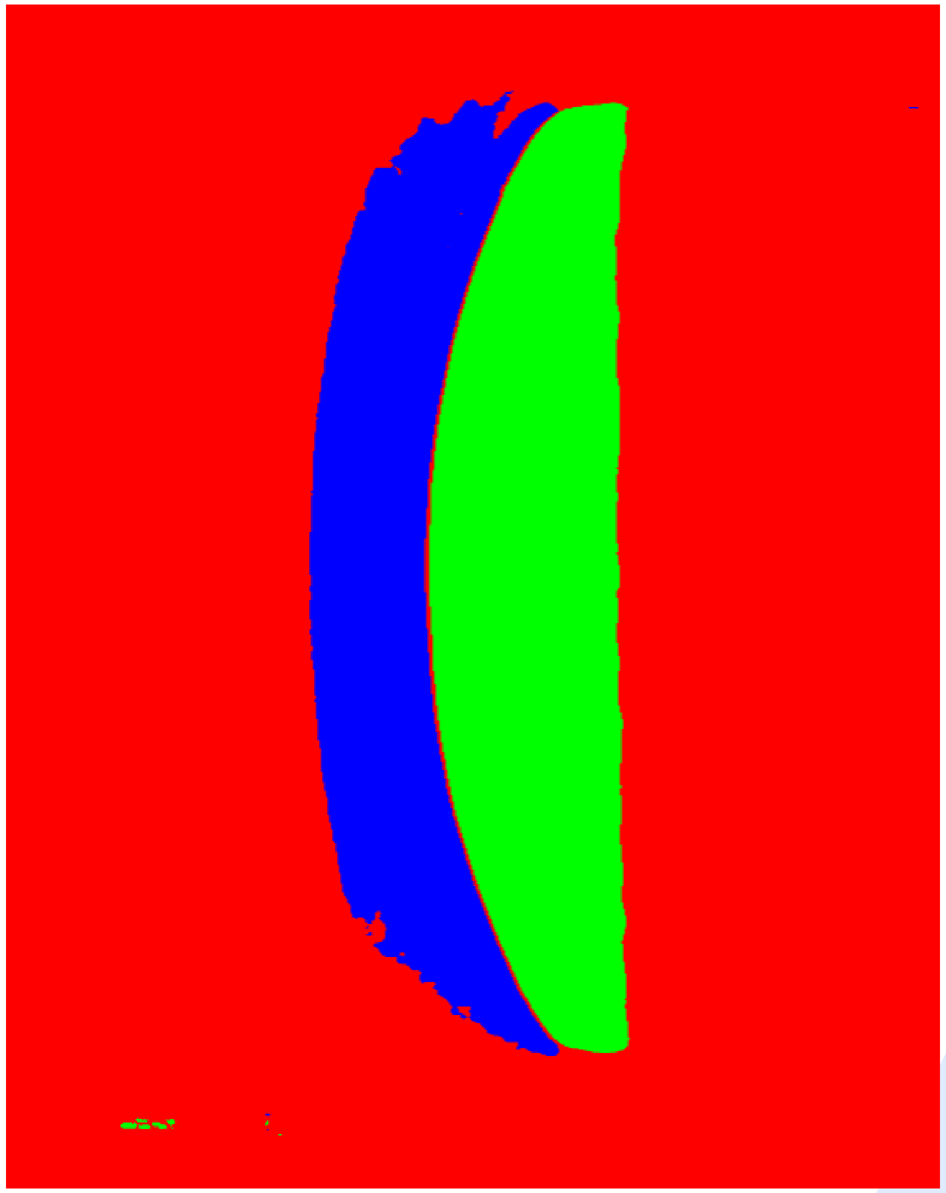
**arcjetCV** uses a CNN with a VGG16-UNet architecture to classify a given image frame into 3 classes:

- material sample
- shock
- background

This segmentation is then passed through simple image processing methods to extract the leading edge of the sample.

# CNN - Results



Shock

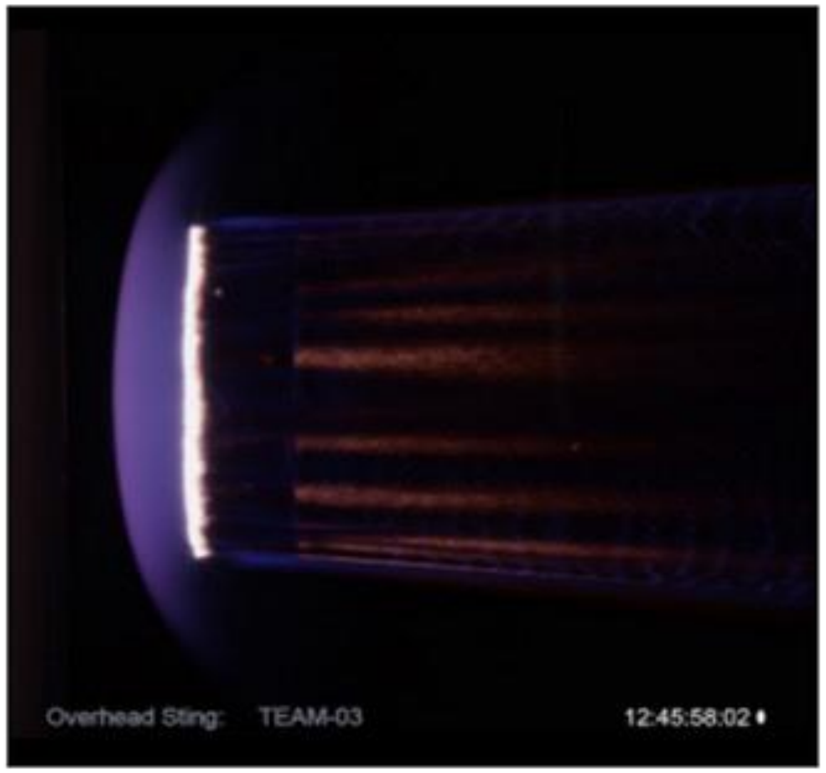
Sample

Background

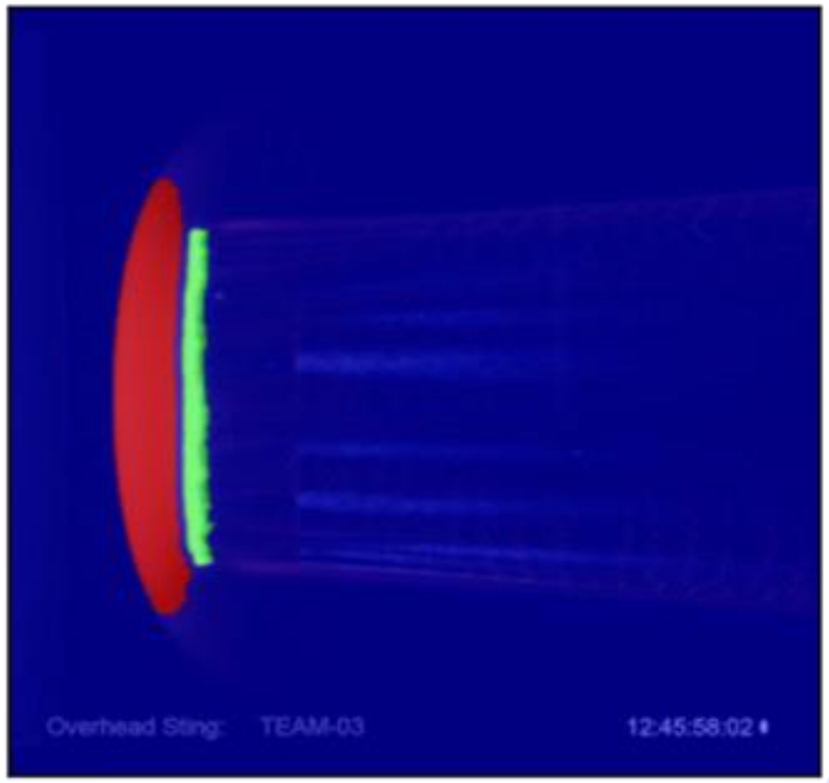
# CNN - Training

**Data labeling:** Manually segmented ~800 representative frames. This was done using the GrabCut algorithm to reduce manual labor & ensure consistent results.

Original Frame




Training mask




Training Data Augmentation

Increased training dataset by changing the orientation and the position of the samples

# CNN - Accuracy

## VGG16- UNet

**F1 score:** $$F1 = \frac{2tp}{2tp + fp + fn}$$

tp = true positive
fp = false positive
fn = false negative

- Sample: 0.9494
- Shock: 0.9789

**Accuracy:** 0.9943

The VGG16-UNet architecture was chosen because it had the best performance out of the available encoder-decoder architectures tested

(PSPNet, ResNet, UNet, miniUNet, SegNet, FCN, MobileNet, etc.)

These parameters represent the segmentation accuracy of the model predictions relative to validation frames not included in the training set

# Decision Tree

Decision tree = many if-statements.

This model was developed to provide a **faster segmentation model** suitable for real-time feedback applications. It uses the HSV (hue, saturation, value) parameter space instead of RGB since the separation between target classes (sample, shock, background) is larger.

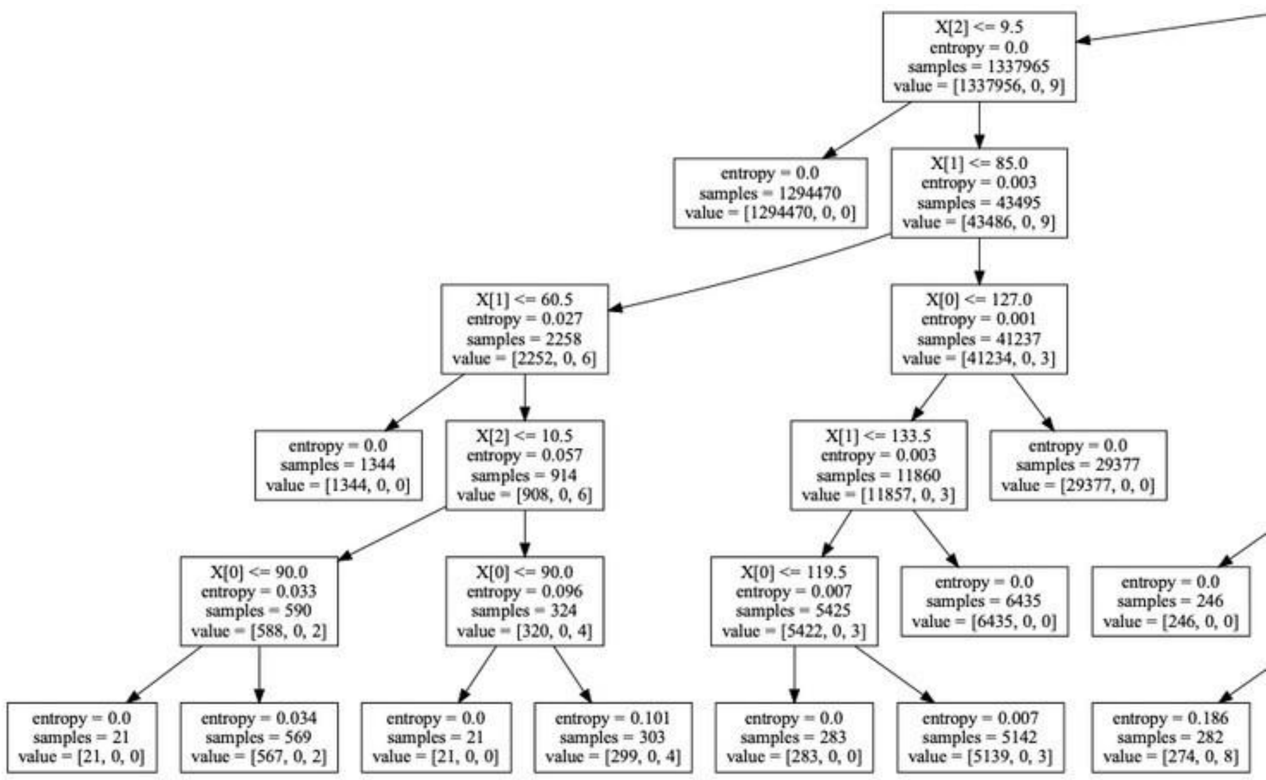
**Decision Tree parameters:**

- Minimum number of samples required to split an internal node = 500
- Depth of the tree = 10
- Input params: HSV of single pixel
- Output: Sample, Shock, Background classification

**Accuracy: 0.93573**

**Applications**:

Realtime feedback for arcjet arm positioning
Realtime feedback for operators/customers

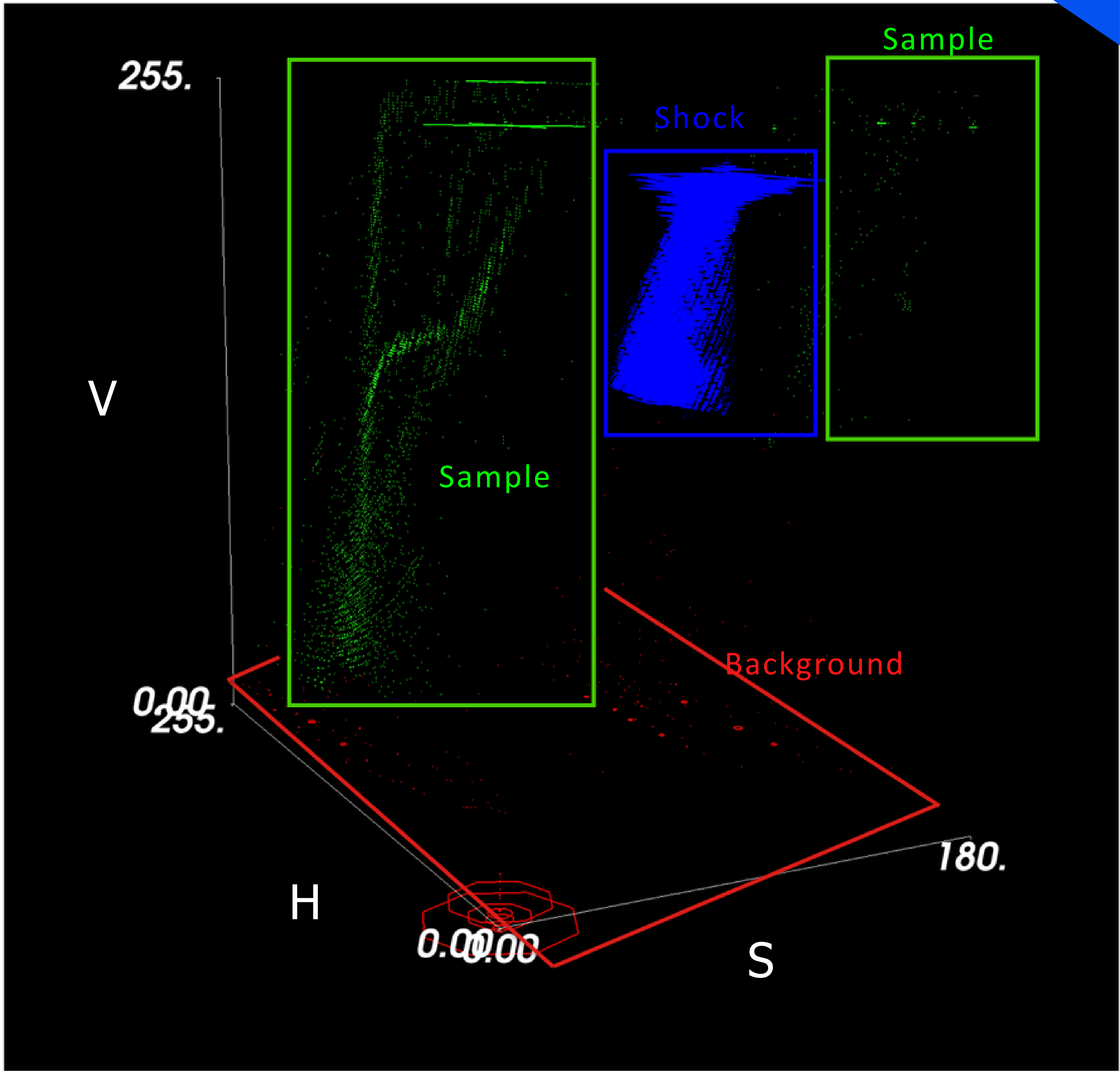


# Decision Tree- Accuracy

### Why a decision tree ?

| Model | CNN | Decision Tree |
|---|---|---|
| Performance | 1Hz | 20Hz |
| Accuracy | 0.9943 | 0.93573 |
| Conclusion | Slow but highest accuracy | Fast with good accuracy |

The decision tree works well because the different classes are mostly separated in HSV space




3D Histogram of Video Pixels

# Time Segmentation

**Typical video:**

25 MINUTES @240 FPS

50 GB, 360k frames

14 insertion clips to manually extract

4 clips are < 3 sec each


MANUALLY CROPPING EACH SAMPLE INSERTION IS **HARD WORK**

Only need to process ~2% of raw video

# Time Segmentation

**Goal:** Train a CNN to find clips of interest within a video

Reduce video data to a simple 1D metric (frame intensity)

Segment normalized 1D intensity data

→ **Problem:**

Integral/derivative thresholding is not sufficiently robust

Not enough data to train CNNs

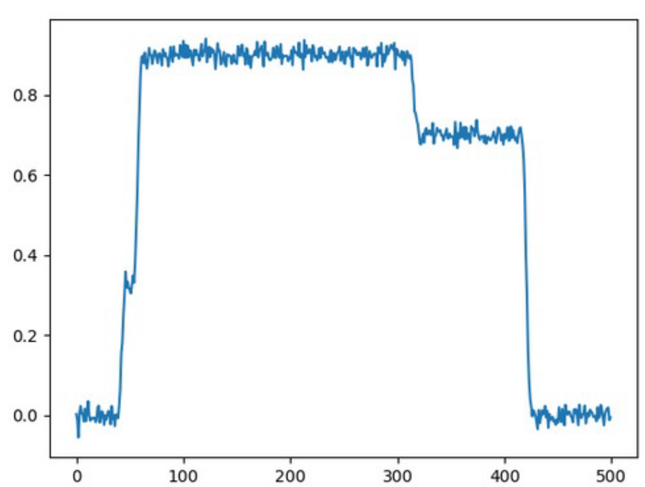
→ **Solution:**

Train CNN with artificial 1D data over a wide range of S/N

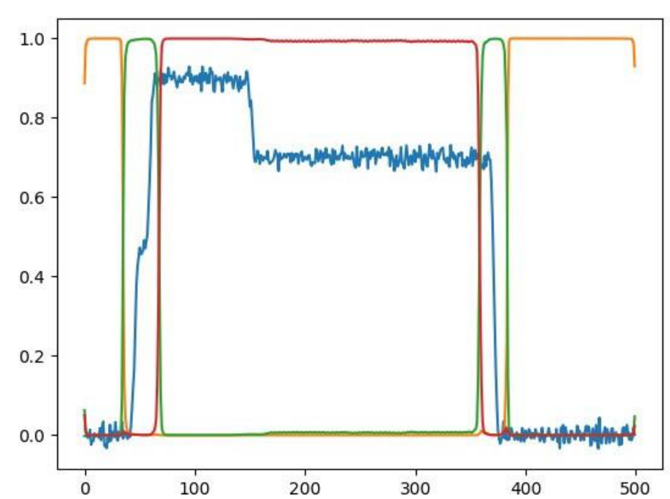
Sample in the frame

No sample

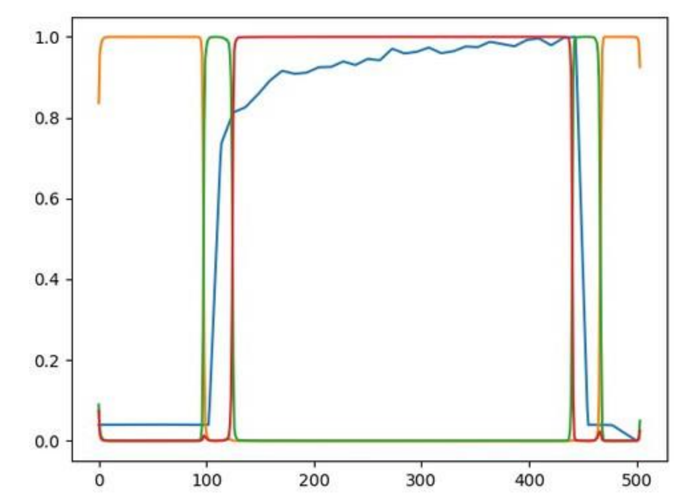
Transitional area




Artificial signal




Training data segmentation

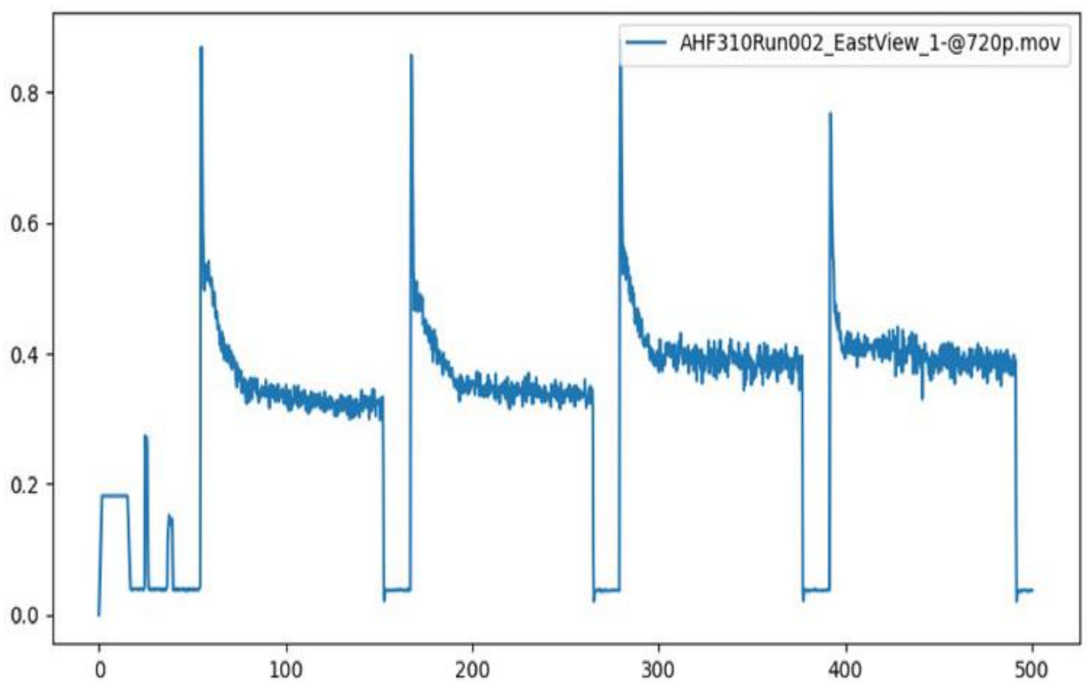


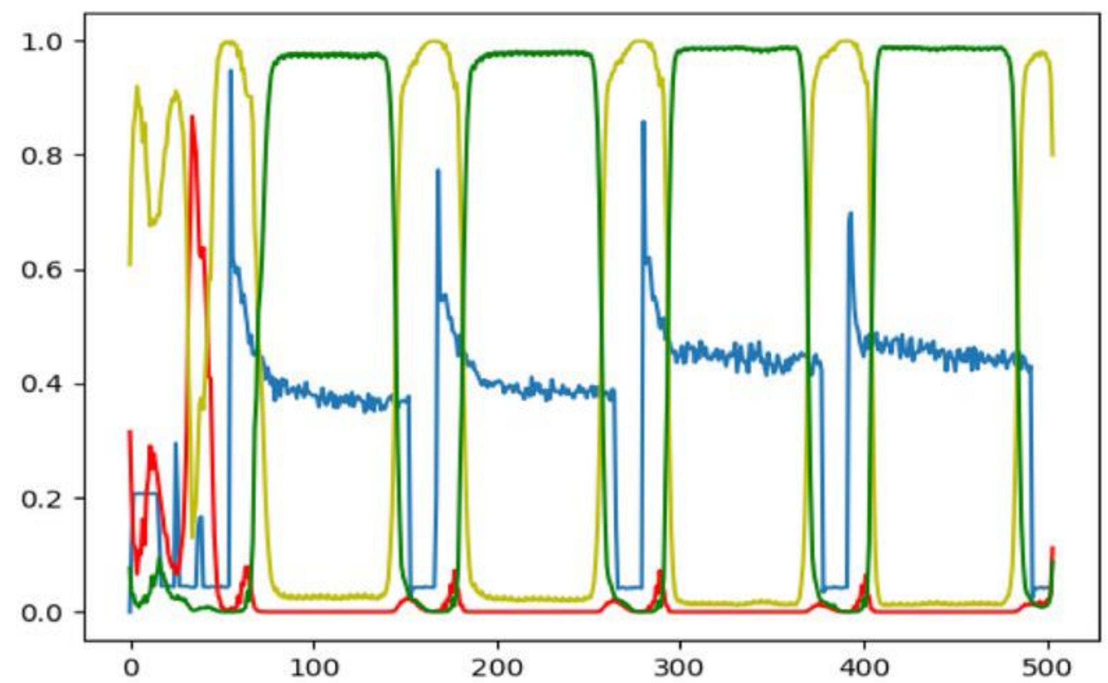

Signal prediction

# Time Segmentation cont’d

Unique features:

- Operates on arbitrary length sequences
- Continuous classification
- Enables multiple events to be parsed even with highly variable data (noise, spikes, changing thresholds, etc.)




Sample in the frame

Transitional area

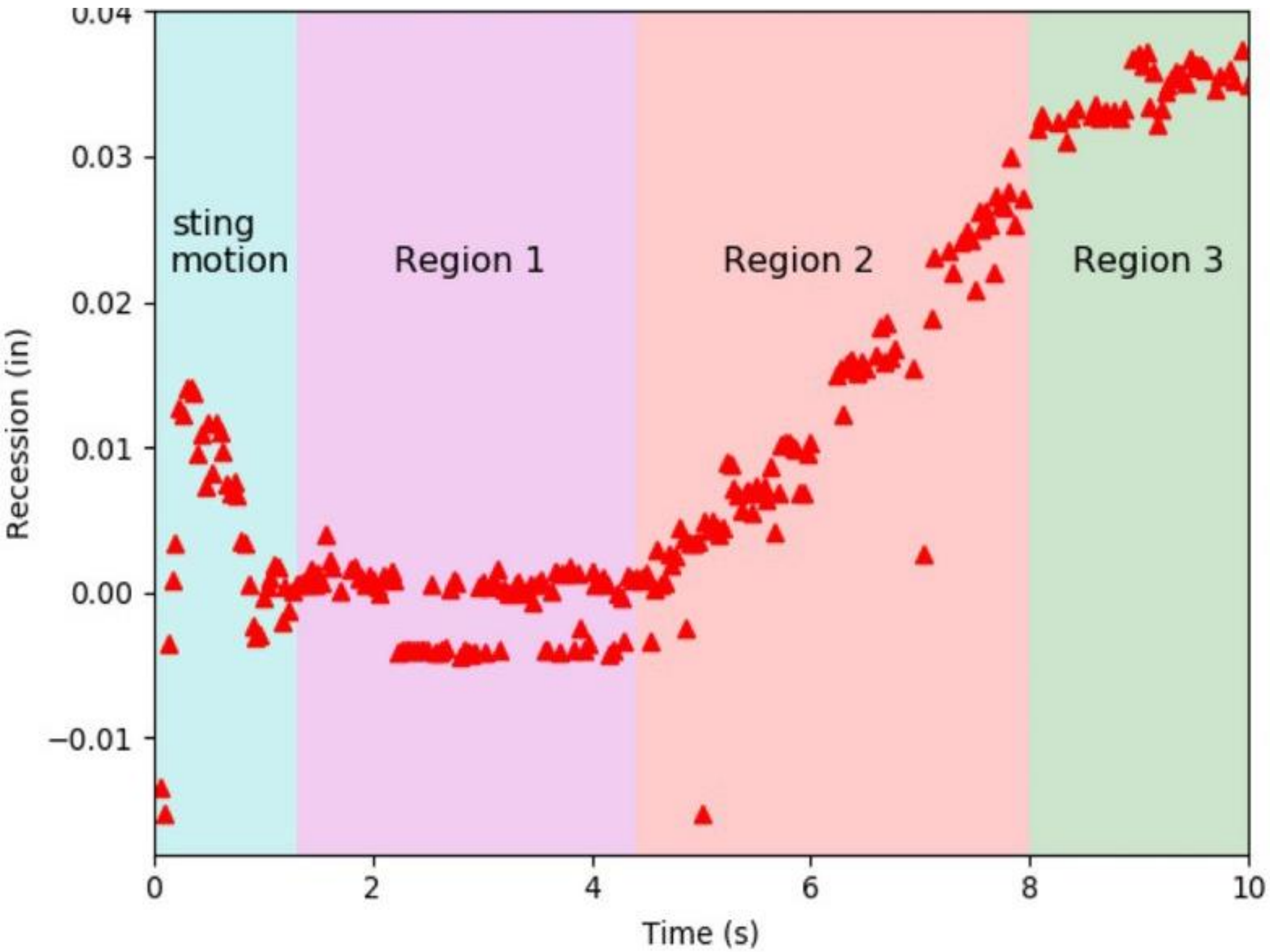
No Sample

Note: this model was originally developed/trained for segmentation of voltage traces but generalizes well to video segmentation. Paper in process on 1D CNNs for generalized time segmentation.

# Discoveries

1. Non-linear recession
2. Changing shock standoff
3. Shape change

# Non-linear Recession




**Before arcjetCV:**
Recession assumed linear

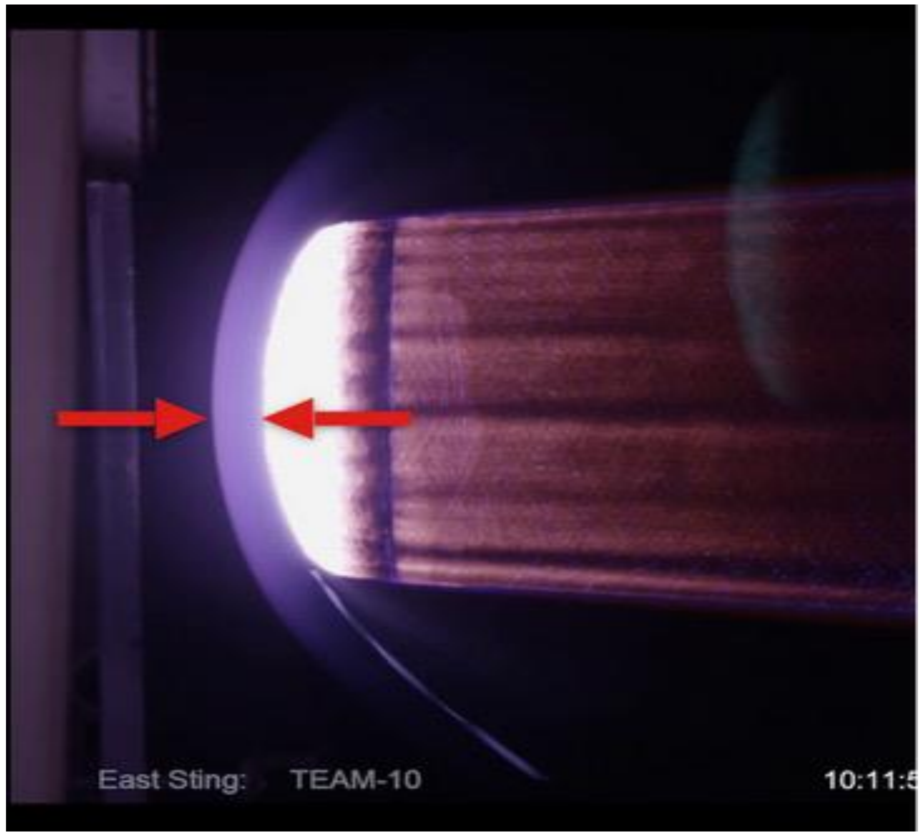
**With arcjetCV:** 3 Regions:

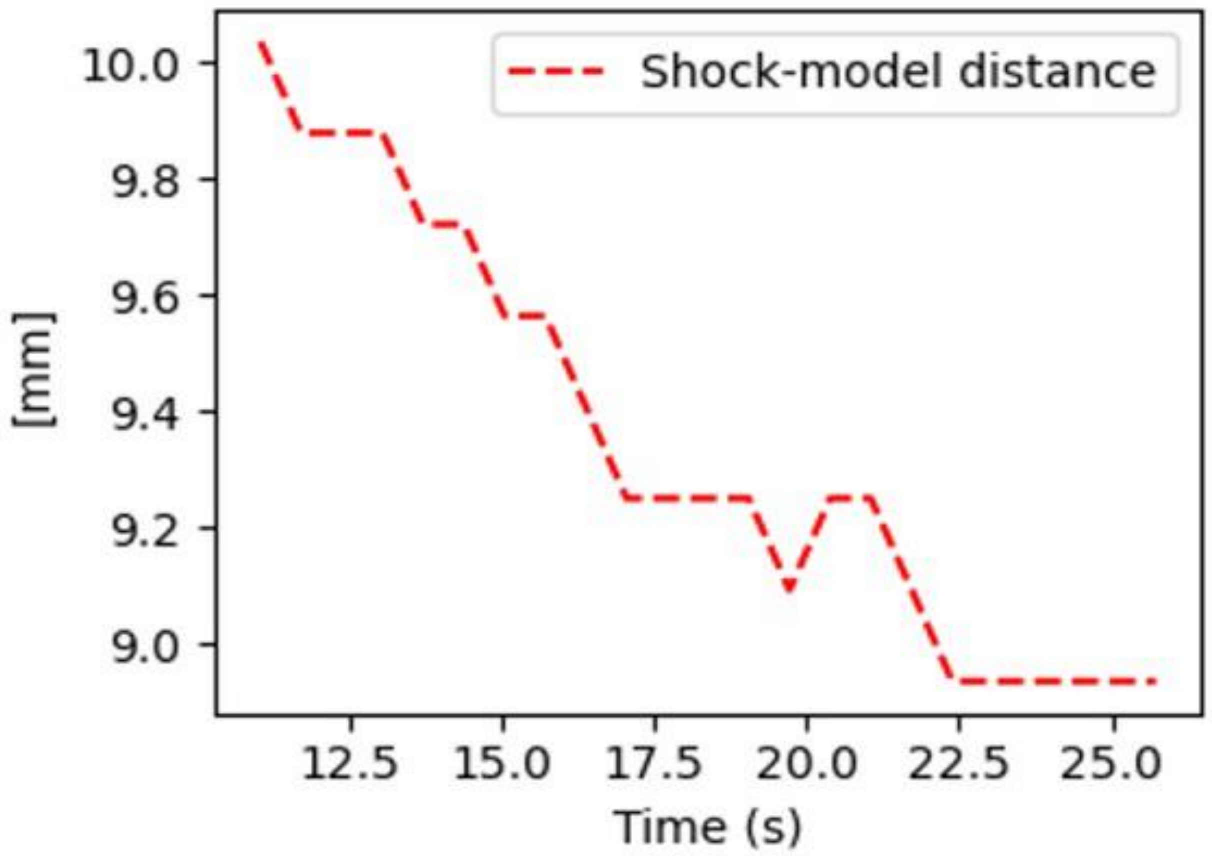
- *Region 1:* No recession
- *Region 2:* Constant rate
- *Region 3:* reduced rate

# Changing Shock Standoff

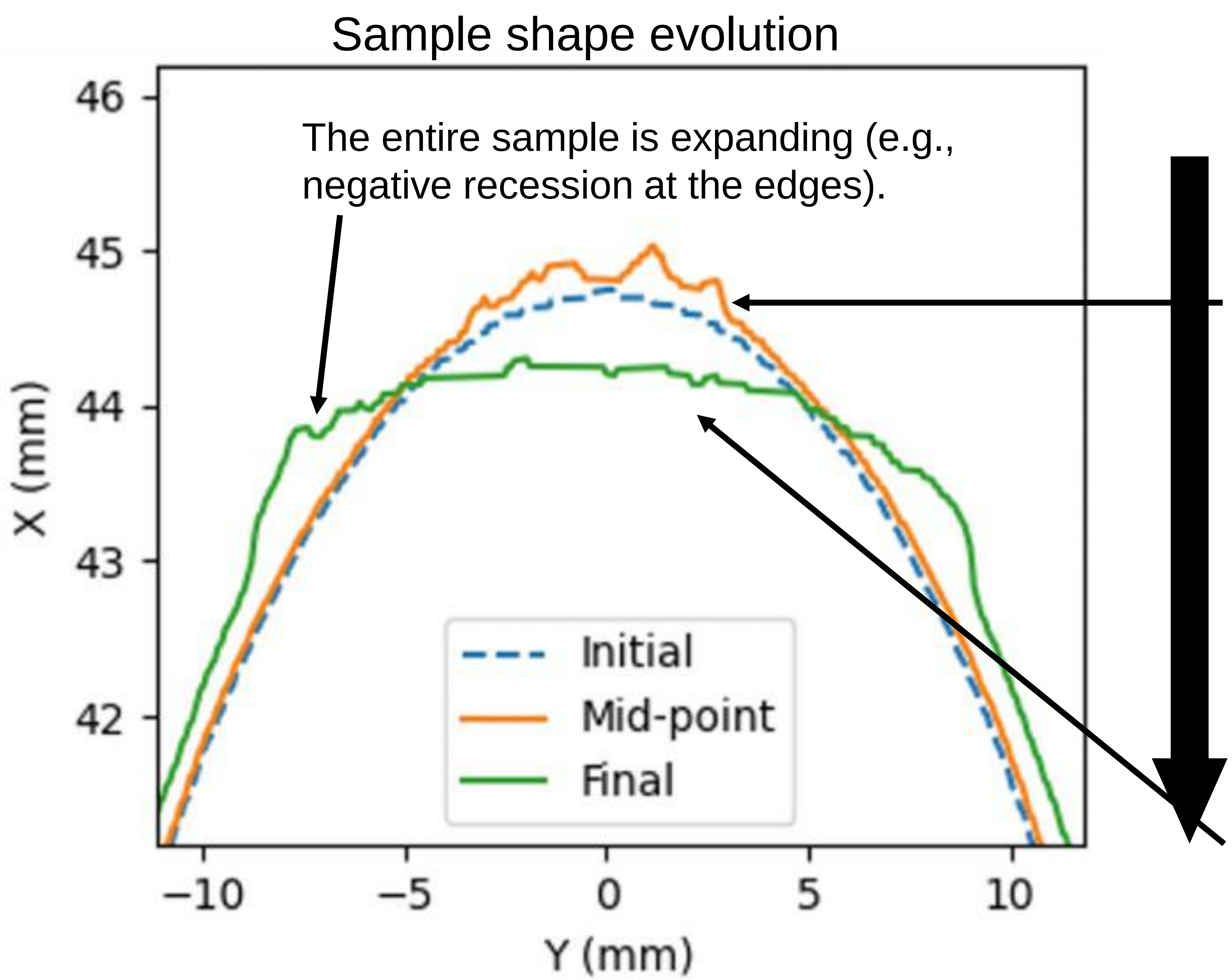
- Validation metric for coupled aerothermal + radiation simulations
- Changing shock standoff distance indicates that the pressure at the sample surface is changing with time

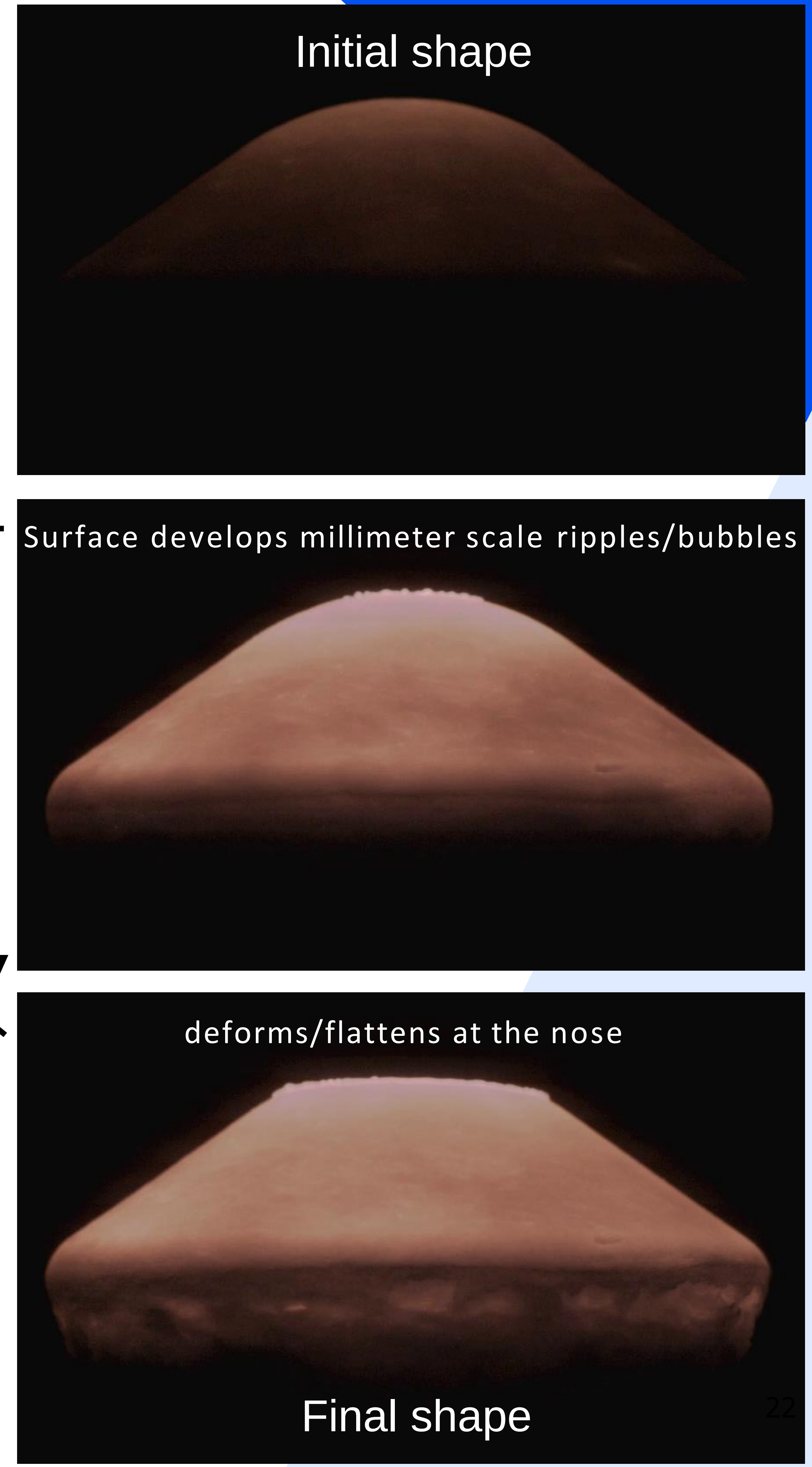
→ Aerothermal conditions are changing
→ The sample is becoming more porous
→ Pyrolysis gas pressure is changing



# Shape change



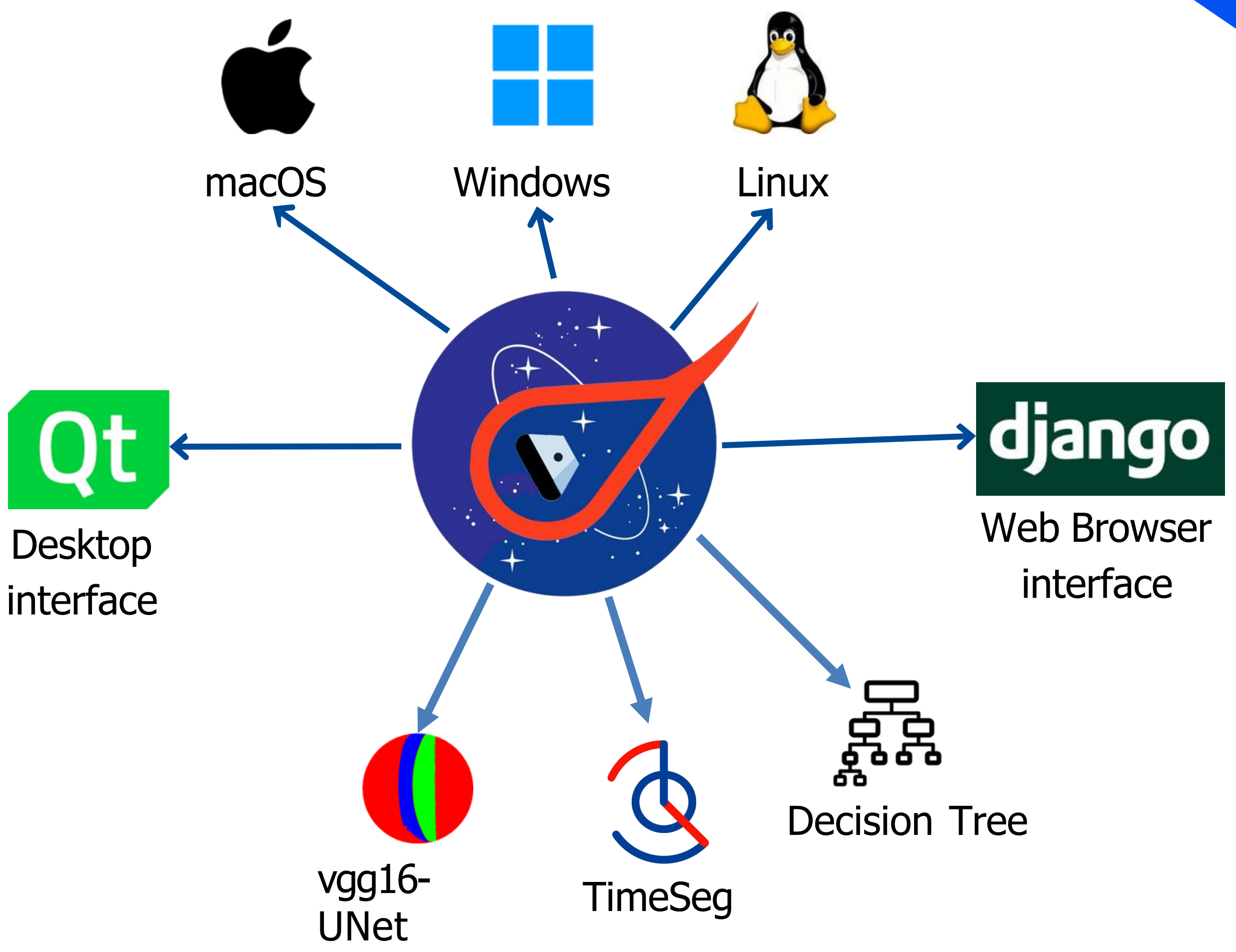




→ Provide the first validation data for new 3D material simulations, tracking melt flow

# Summary

## arcjetCV

- 3 ML models
- 2 interfaces
- 3 operating systems




## Enables

- Time-resolved recession
- Characterize swelling/shrinkage/melt flow
- Time resolved shock standoff
- 2D recession validation

## NASA Users

- Ames arcjet facilities
- MSR-EES
- Orion

→ Better mission planning and risk management

# Future Work = More Users?

We are in the process of releasing arcjetCV as an open-source software

Development of several new features and improved user interfaces is ongoing

Please contact me if interested in using arcjetCV

magnus.haw@nasa.gov

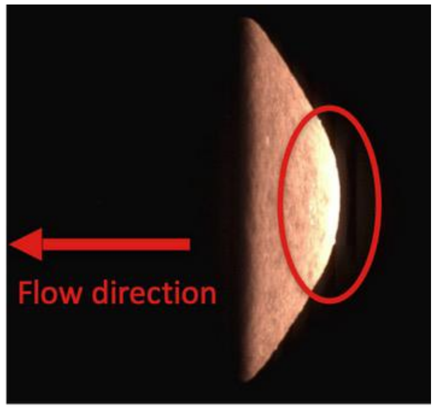
NASA Ames Research Center
Thermal Protection Materials Branch


Thank you
for your
attention !
magnus.haw@nasa.gov
alex@flying-squirrel.space

# AutoCrop, AutoDirection, AutoOutlierRemoval

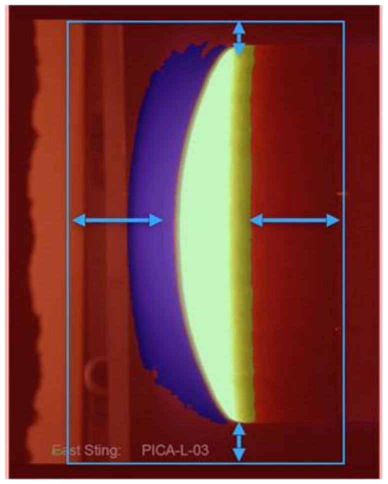
**1) AutoDirection**

- Calculates the weighted average location of frame intensity
- Depending on the location relative to image center, the flow direction is inferred




**2) AutoCrop**

- Segment a frame in the middle of the video
- Get the extrema pixel location and bounding box
- Set the crop window by adding a percentage to the minimum bounding box width and height




**3) Outlier Removal**

- Mask extracted time series using LocalOutlierFactor (LOF) method from scikit-learn
- LOF is an unsupervised algorithm based on the deviation from nearest neighbors
- This masks significant outliers enabling the time-series linear fitting to be more robust

# Web Interface Development

**Problem:**

- Difficult to provide support for all operating systems
- Time consuming to update software on many machines individually

**Solution:**

Setup arcjetCV on a secure local server

-Use SSO (NASA Launchpad for authentication)

-Access controlled via NAMS requests

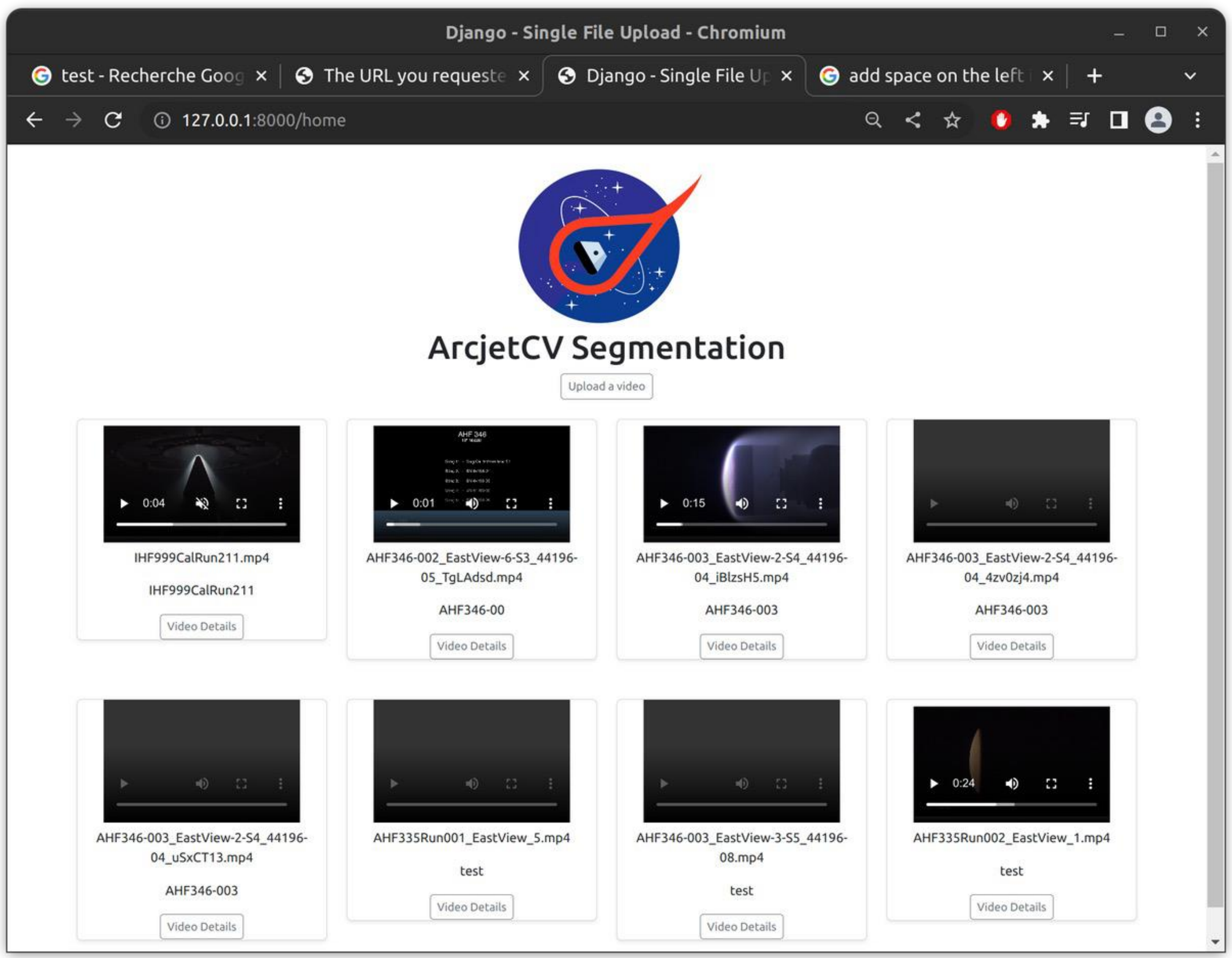
-Limited upload file size (<1 GB)

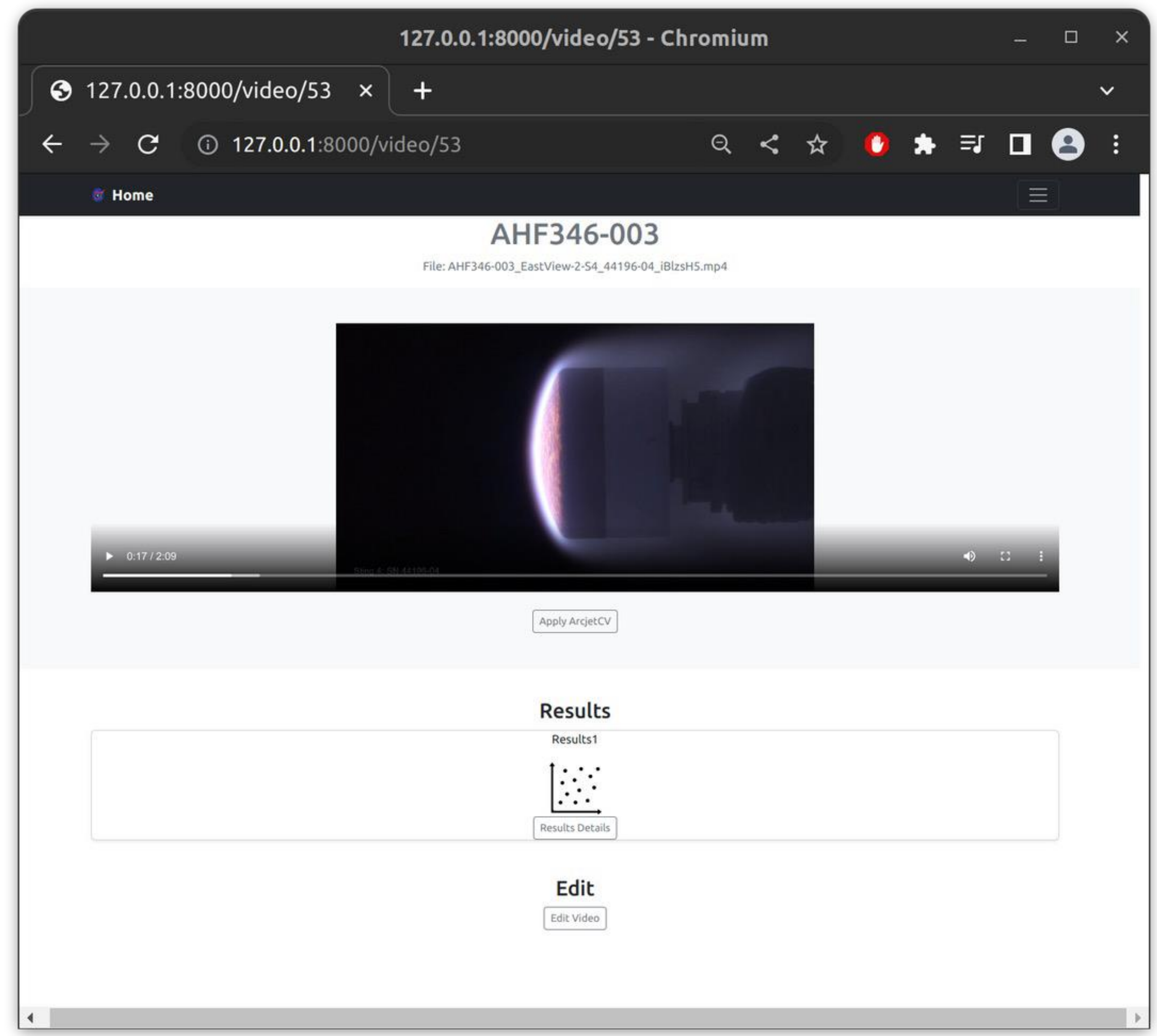
-Limited concurrent users

-Significantly harder to release as a software

# Web Browser Interface

## Setup arcjetCV on a secure local server



# Web Browser Interface

## Setup arcjetCV on a secure local server