Utilizing Generative Adversarial Networks to Produce Airfoil Geometries

Zachary Eckley
Faculty Advisor – Khem Poudel

Background

- A generative adversarial network (GAN) uses a series of convolutional layers to create new instances of data that closely resemble real data from the training set.
- The model is made up of two submodels, the generator and the discriminator.
  - The generator creates new data.
  - The discriminator determines if the data is real.
- Over several iterations, or epochs, the models improve at their respective roles.
- In this study, a GAN was applied to generate unique airfoil geometries based on a set of airfoil performance data.
- Typically, airfoils are designed using CFD, optimization algorithms, and wind tunnel testing. These methods can be time consuming and costly.
- The objective was to create a faster, cheaper alternative to current industry design methods

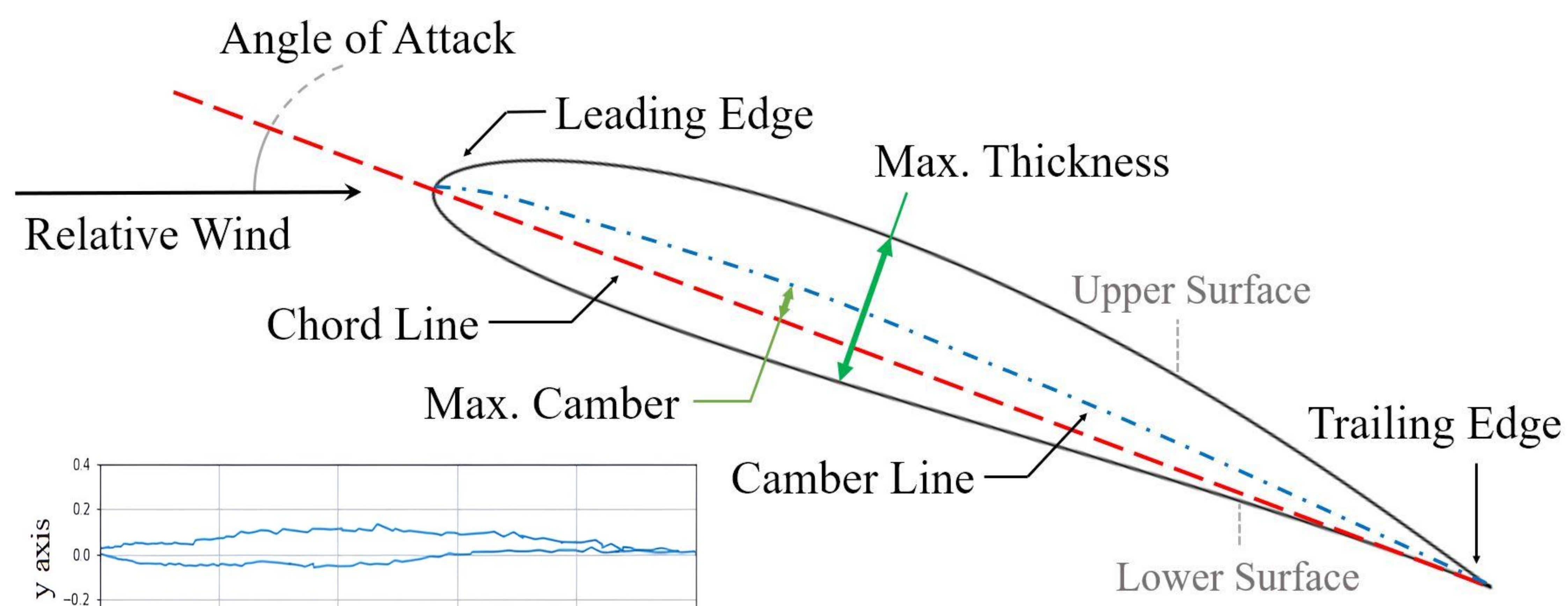
Model Structure

Generator

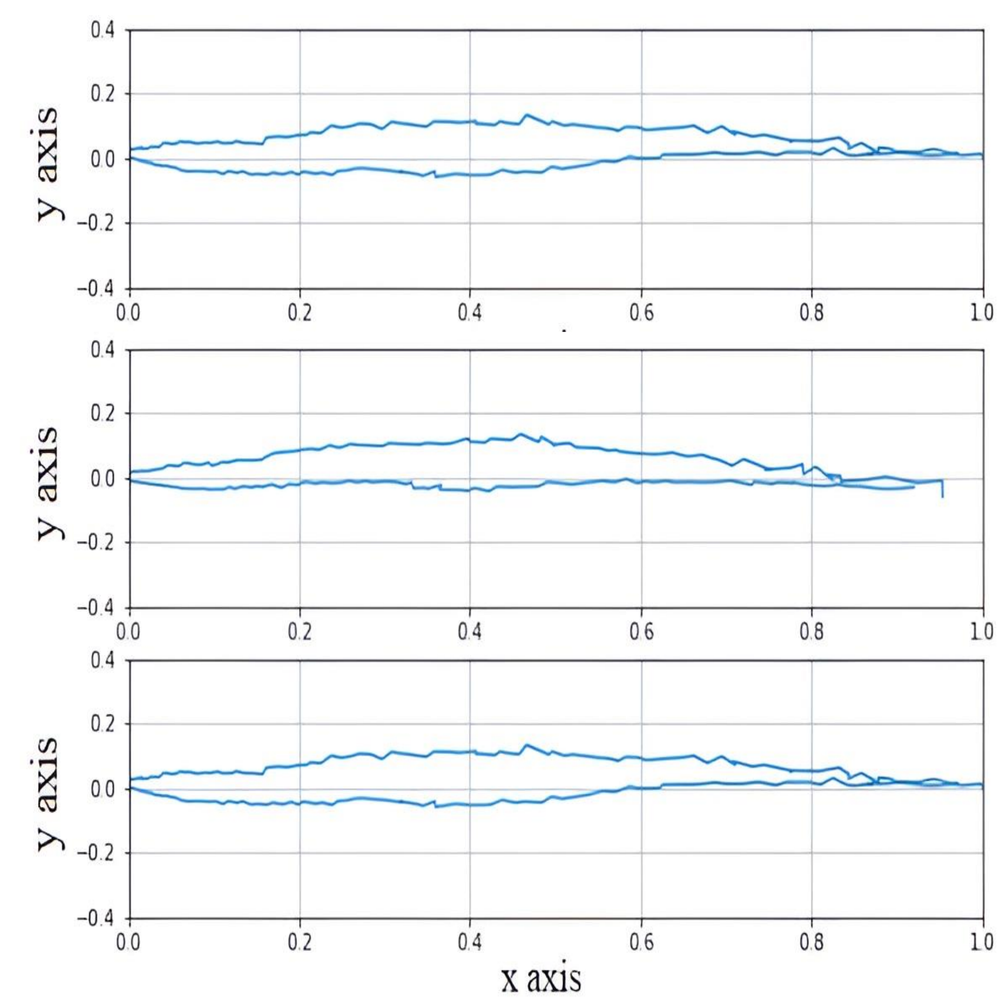
Layer	Filters	Kernel Size	Stride	Output Shape
Input Layer	—	—	—	(100,1)
Dense	—	—	—	(100,1)
Spectral Normalization	—	—	—	—
Reshape	—	—	—	(2,25,1024)
2D Deconvolution	1024	(2,4)	(1,2)	(2,50,1024)
Spectral Normalization	—	—	—	—
2D Deconvolution	512	(2,4)	(1,2)	(2,100,512)
Spectral Normalization	—	—	—	—
2D Deonvolution	256	(2,4)	(1,2)	(2,200,256)
Spectral Normalization	—	—	—	—
2D Convolution	1	(2,2)	(1,1)	(2,200,1)

Discriminator

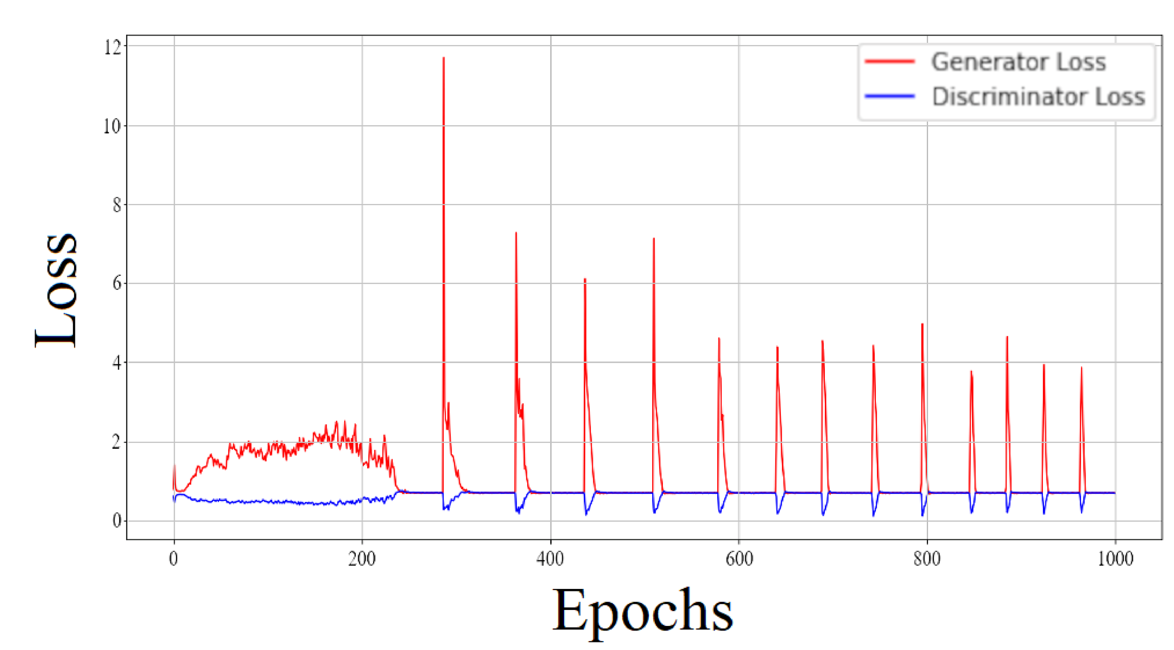
Layer	Filters	Kernel Size	Stride	Output Shape
Input Layer	—	—	—	(2,200,1)
2D Convolution	128	(2,4)	(2,2)	(2,100,128)
Batch Normalization	—	—	—	—
2D Convolution	128	(2,4)	(2,2)	(2,50,128)
Batch Normalization	—	—	—	—
Flatten	—	—	—	(12800,1)
Dense	—	—	—	Scalar



Airfoil cross-section with labeled geometry



Output from model after 1000 epochs



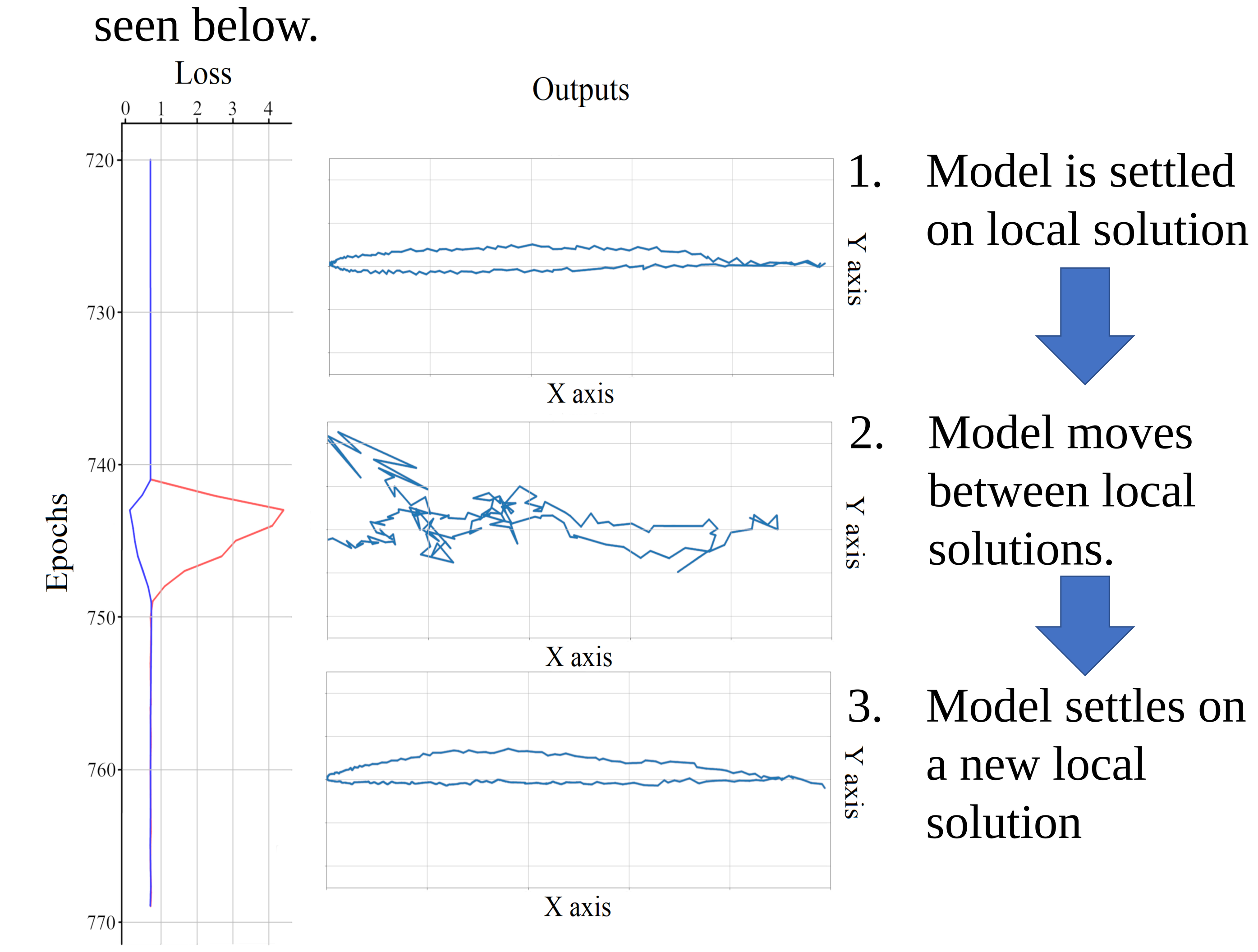
Loss curve from model over 1000 epochs

Methodology

1. Airfoil Coordinates were collected from a database created by University of Illinois Urbana-Champaign (UIUC). Bezier curves were fit to the dataset coordinates to interpolate a constant 200 points for each airfoil.
2. The interpolated coordinates were analyzed at 121 Reynolds numbers. At each Reynolds number, several performance characteristics such as lift, drag, and moment are found at several angles of attack.
3. The resulting performance data forms a large matrix [121x201x7], this matrix was passed through an encoder model that reduced the dimensionality without losing information.
4. The reduced performance matrix is then used as input for the generator.
5. The discriminator is given some of the original airfoil coordinates and some of the generated airfoil coordinates and determines which are real and which are fake.
6. The loss is calculated for the discriminator based on its accuracy. The lower the loss the better. Incorrect identifications of real/fake airfoil coordinates increase the loss score.
7. The model was trained for several epochs to continuously lower the loss. Training was stopped after 1000 epochs or when the loss converged on a value.
8. Adjustments were made to both model's structure to improve the results.

Results

- Outputs resemble airfoil geometry. Smoothness could be fixed with post-processing.
- Most of the outputs look very similar
- The generator loss works through a transient period before equalizing with the discriminator.
- Once the generator and discriminator equalize massive spikes in the generator's loss begin to occur.
- Further analysis of these spikes resulted in the conclusion seen below.



Output of model at each epoch compared to the loss from model at each epoch

- Generator suffers from mode collapse, a common problem with GANs where the outputs represent a small subset of the original dataset.

Looking Ahead

1. Use Wasserstein loss function rather than binary-cross-entropy.
2. Use a conditional GAN to provide the performance data to the generator rather than the standard structure.
3. Have the generator output an image of the airfoil coordinates rather than numeric values
4. Additional layers could be added to the generator and discriminator, but more computing power would be required.