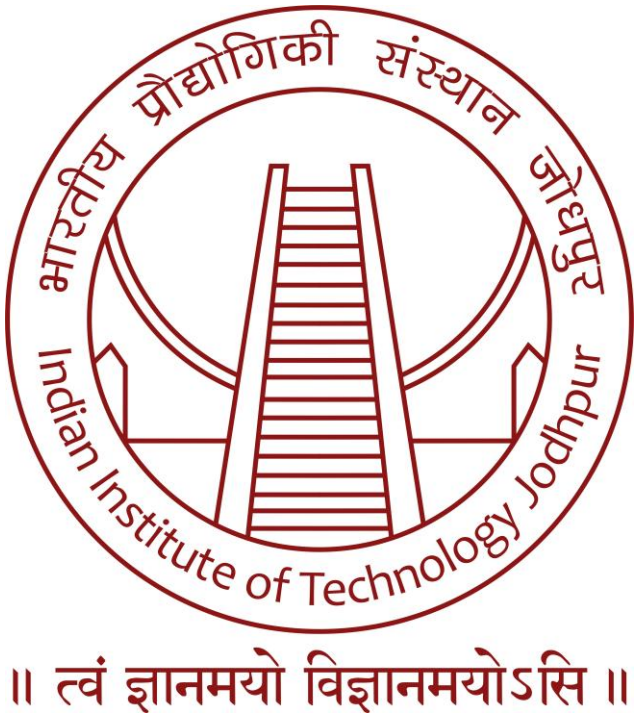

Figure 1: IIT Jodhpur

MEL2040 Project Data-Driven Analysis of Fluid Flows

by

Gyanendra Bhardwaj

Supervised by Professor. HARSHAL AKOLEKAR

Submitted April 15, 2024

Student ID: B22ME021, Email: b22me021@iitj.ac.in

Table of Contents

Main Content

1 Review of Machine Learning in Fluids

1.1 Introduction

Humans are the only known species to use the Fluid mechanics equations for fluid manipulation. Our ancestors used intuition to build fluid-based systems like dams and mills, but the last century's scientific advancements have significantly improved these designs. Despite this progress, fluid dynamics' complexity often hinders real-time control. Now, as we harness machine learning and data optimization, we're relearning from experience, opening new frontiers in fluid mechanics.

1.2 Advances of machine learning in the domain of fluid mechanics.

Flood Prediction Using Machine Learning Models

Introduction

The paper in delves into the critical role of machine learning (ML) models in the domain of flood prediction. It underscores the pivotal contributions these models make towards mitigating risks associated with floods and enhancing the efficacy of disaster management systems. The primary objective of this paper is to conduct a thorough evaluation of cutting-edge ML models that are currently employed in the field of flood forecasting. The paper aims to shed light on the most effective models, providing valuable insights into their suitability for various predictive tasks.

Methodology

The literature review meticulously examines a spectrum of ML models, including Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and a variety of hybrid models. The analysis is comprehensive, focusing on the robustness, accuracy, efficiency, and processing speed of these models. The paper recognizes the significant strides made in the realm of flood prediction, attributing these advancements to innovative strategies such as the hybridization of models, data decomposition techniques, the ensemble of algorithms, and the optimization of existing models. These methodologies have been instrumental in refining the predictive capabilities of ML models, thereby contributing to more reliable flood forecasting.

Conclusion

The paper concludes that hybrid ML methods, data enrichment, and ensemble models show promise for enhancing flood prediction accuracy. Future Direction: It suggests that further research is needed to explore advanced hybrid models and address the generalization problem in ML applications for flood prediction. This review encapsulates the paper's comprehensive analysis of ML models in the context of flood prediction

Machine-learning based optimization of a biomimicked herringbone microstructure for superior aerodynamic performance

Introduction

The paper delves into a fascinating intersection of biomimicry and aerodynamics. Specifically, it explores the optimization of a biomimicked herringbone microstructure, drawing inspiration from nature's intricate designs—particularly bird feathers. The motivation behind this research lies in the quest for improved aerodynamic performance, a critical factor in various engineering applications. Birds have evolved over millions of years to achieve remarkable flight efficiency. Their feathers, with their intricate patterns and microstructures, play a pivotal role in reducing drag and enhancing lift. By mimicking these natural designs, engineers seek to unlock similar benefits for man-made systems, such as aircraft, automobiles, and marine vessels.

Main Body

Integration of Disciplines The heart of this study lies in the seamless integration of three distinct disciplines: computational fluid dynamics (CFD), machine learning (ML), and biomimicry. Each discipline contributes unique insights and tools to address the challenge of optimizing herringbone riblet patterns for aerodynamic gains.

Computational Fluid Dynamics (CFD): CFD provides a virtual laboratory for analyzing fluid flow around complex geometries. Researchers simulate airflow over the biomimicked herringbone microstructure, capturing intricate details such as boundary layers, vortices, and pressure distributions. Through CFD simulations, they quantify drag coefficients, lift forces, and other aerodynamic parameters. These simulations serve as a bridge between theory and practical design. Machine Learning (ML): ML algorithms play a crucial role in predicting and optimizing aerodynamic performance. Researchers train ML models using historical data from CFD simulations and experimental tests. These models learn patterns, correlations, and nonlinear relationships between design parameters (such as riblet spacing, angle, and depth) and aerodynamic outcomes. By leveraging ML, engineers explore vast design spaces efficiently, identifying optimal configurations that minimize drag and maximize lift. Biomimicry: Nature's blueprints guide the creation of the herringbone microstructure. Researchers meticulously analyze bird feathers, unraveling their secrets. The herringbone pattern—a series of overlapping V-shaped grooves—proves particularly intriguing. It disrupts laminar flow, reducing skin friction and delaying flow separation. Biomimicry informs the design process, ensuring that the artificial microstructure closely resembles its natural counterpart.

Methodology and Results

Design Creation: Researchers generate various herringbone riblet patterns, adjusting parameters such as groove spacing, angle, and depth. These designs span a wide range, from subtle variations to radical departures from traditional riblet geometries. CFD Simulations: The herringbone microstructures undergo rigorous CFD simulations. Researchers analyze flow patterns, pressure gradients, and drag forces. They quantify the impact of different design

choices on aerodynamic performance. Iterative simulations refine the designs, converging toward optimal configurations. Machine Learning Predictions: ML models predict drag coefficients based on input parameters (e.g., riblet dimensions, Reynolds number, and flow conditions). Researchers validate these predictions against experimental data and refine the models iteratively. ML accelerates the exploration of design space, guiding engineers toward promising solutions.

Conclusion

The optimized herringbone microstructure demonstrates reduced drag, making it suitable for aerospace, automotive, and marine applications. Imagine airplanes gliding through the sky with less resistance, electric cars achieving longer ranges, and ships navigating turbulent waters more efficiently. The synergy of CFD, ML, and biomimicry exemplifies the power of interdisciplinary collaboration. As we look ahead, further research will refine these designs, address real-world constraints, and propel us toward a more sustainable and streamlined transportation future. In summary, this paper not only celebrates nature's ingenuity but also charts a course for engineering innovations inspired by the elegance of bird feathers. By combining science, computation, and biomimicry, we inch closer to a world where our machines move with the grace of avian flight

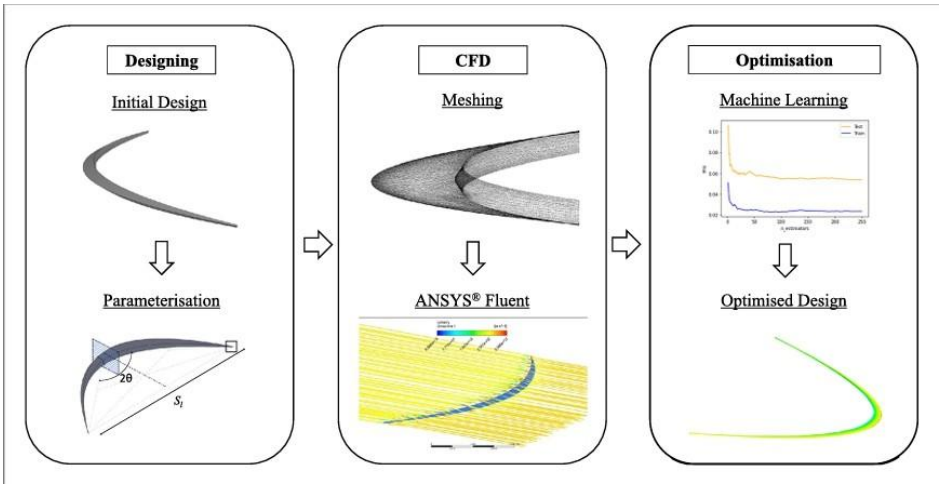

Figure 2: Workflow of herringbone riblet microstructure optimization using machine learning.

2 Any New Ideas

How AI/ML tackles shipping chaos and prevents accidents Machine Learning (ML) is pivotal in mitigating the shipping industry's challenges. Implementing ML models enables the analysis of demand and supply fluctuations. The infamous Suez Canal obstruction by the Ever Given ship, partly due to strong winds, resulted in significant losses. Such incidents can be minimized through the deployment of onboard navigation sensors and high-resolution cameras, coupled with specialized AI algorithms. These technologies empower the crew to make swift, informed decisions, enhancing their ability to observe and address emerging issues promptly. This advanced level of situational awareness is key to reducing the approximately 4,000 maritime accidents that happen each year, often due to poor situational awareness in busy areas, inadequate office insight into misses and risk patterns, and a lack of comprehensive data for managing potential incidents.

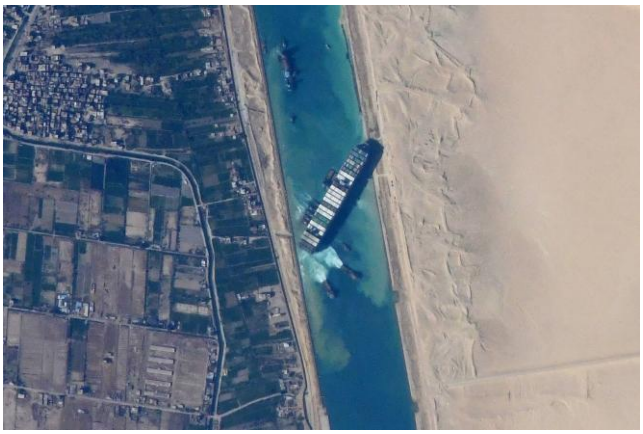

Figure 3: Suez Canal blocked by the ship **Evergreen**

Microfluidic Device Design: Microfluidics is a rapidly growing field due to its ability to perform many laboratory functions more efficiently and with less reagent than traditional methods. The technology's potential for innovation in healthcare, research, and industry is immense, making it a key area of interest in modern science and engineering. The application of machine learning could revolutionize the design process of microfluidic devices. Machine learning has the potential to transform the design process for microfluidic devices These devices have a broad spectrum of uses, including but not limited to, medical diagnostics and chemical synthesis. By training machine learning algorithms on datasets from past designs, we could forecast the efficacy of new designs. This approach could not only expedite the design process but also enhance the functionality of these devices.

New Media Art Created as a Visualization of Fluid Dynamics Fluid mechanics in Art: Unleashing Unpredictability Our exploration of media art, driven by technology, now delves into fluid dynamics. This captivating field, also known as "fluid mechanics," has witnessed extensive research. While some fluid motions possess inherent beauty, there's an intriguing niche: the "visualization of fluid motion." However, existing results predominantly

showcase stable behaviors, overlooking the unpredictable and chaotic aspects. These elusive dynamics hold immense artistic potential, especially within image sciences. When dealing with images, machine learning offers boundless possibilities. Consider the once-unimaginable task of transforming a static image of a famous artist's waterfall into a mesmerizing movie. This fusion of fluid dynamics, art, and ML opens new creative vistas

Figure 4: Van Gogh, Vincent . The Starry Night. 1889, oil on canvas, Museum of Modern Art, New York.

Fluid Dynamics and Combustion Stability Prediction in Aviation Engines One fascinating area of research in machine learning (ML) and artificial intelligence (AI) is the prediction of precursors related to the stability of engine combustion. Specifically, this field focuses on thermo-acoustic instabilities occurring within aviation engines. The intricate interplay between fluid scales, acoustic scales, and combustion scales significantly influences engine dynamics. However, predicting these interactions remains a complex challenge. Moreover, similar techniques can be applied to predict precursors for transition phenomena.

Figure 5: Aviation Engine

3 Proper Orthogonal Decomposition

[Project Link]

3.1 Image Generation

- The given 24 fps input video was sliced into frames using Open-CV.
- While converting the video, some frames were skipped in a way reducing fps to reduce computation costs. The number of total frames were reduced from 750 to 188.
- A function then flattens each image and concatenates them into 1 dataset. The final dataset shape was (1103064, 188, 3).

Brief explationation of execution of image generation

An input video, originally running at 24 frames per second (fps), was dissected into individual frames using Open-CV, a popular computer vision library. The objective was to transform the video data into a more manageable form for further processing. However, to optimize computational resources, a strategic decision was made during the conversion process. Instead of using every frame, some were deliberately skipped, effectively lowering the fps. This selective frame extraction led to a significant reduction in the total frame count, from 750 to a more manageable 188.

Following the frame extraction, each frame was subjected to a flattening operation. This process transformed the 2D image data into a 1D array, simplifying the data structure and making it more suitable for the subsequent steps.

The final stage involved concatenating these flattened frames into a single dataset. This operation effectively combined the data from all the frames, creating a comprehensive representation of the entire video. The resulting dataset had a shape of (1103064, 188, 3), indicating it contained over a million data points, spread across 188 frames, each with three color channels. This compact and efficient representation of the video data is ideal for many machine learning and data analysis tasks.

3.2 Execute POD

- Initially, our data was subjected to scaling using the Standard Scaler function, which standardizes a feature by subtracting the mean and thereafter scaling it to have a variance of one.
- The numpy library was utilized to conduct matrix operations and obtain the Spatial Mode Matrix U, Energy Matrix Σ, and the Transposed Temporal Mode Matrix V^T.
- Once the data has been divided into three matrices, we select the most important values and decrease the size of the Energy Matrix from $m \times n$ to $r \times n$, where r is the rank at which the cumulative variance explained by the eigenvalues exceeds a certain threshold.

- The updated matrices U_r, Σ_r, and V_r^T are multiplied together to obtain the rebuilt matrix A_r.

$$A = U\Sigma V^T = \underbrace{\begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_r\end{bmatrix}}_{\text{Col } A} \underbrace{\begin{bmatrix}\mathbf{u}_{r+1} & \dots & \mathbf{u}_m\end{bmatrix}}_{\text{Nul } A^T} \begin{bmatrix}\sigma_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 & \dots & 0 \\ \dots & & & & & & \\ 0 & 0 & \dots & \sigma_r & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & & & & & & \\ 0 & 0 & \dots & 0 & 0 & \dots & 0\end{bmatrix} \begin{bmatrix}\mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \dots \\ \mathbf{v}_r^T \\ \mathbf{v}_{r+1}^T \\ \dots \\ \mathbf{v}_n^T\end{bmatrix} \begin{matrix}\} \text{Row } A \\ \\ \\ \\ \} \text{Nul } A\end{matrix}$$

As shown in the code snippet below, we have implemented SVD from scratch using the numpy library in python.

```
def SVD(A, frames):
  A_T = np.transpose(A)
  A_2 = np.matmul(A_T, A)
  eigenvalues, V = LA.eig(A_2)
  E = np.zeros((frames,frames))
  for i in range(frames):
    E[i][i] = np.sqrt(eigenvalues[i])
  print(eigenvalues.shape)

  V_T=np.transpose(V)
  V_T_inv=LA.inv(V_T)
  E_inv = LA.inv(E)
  U1 = np.dot(A,V_T_inv)
  print(U1.shape)
  U = np.dot(U1, E_inv)
  return U, E, V_T, eigenvalues

U, E, V_T, eigenvalues = SVD(A, frames)
sorted_eigens = np.argsort(eigenvalues)[::-1]
```

Listing 1: POD Code

3.3 Analyse POD Modes

1. Analysing Energy Matrix Σ

   - The matrix is a diagonal matrix where each element on the diagonal represents the amplitude of the corresponding mode.
   - Figure 6 illustrates that the first modes possess a significant quantity of energy, while the subsequent modes exhibit a negligible energy magnitude. This suggests

that the matrix can be approximated by considering only a limited number of modes.

- The first mode exhibits a significantly high magnitude since it represents the mean value of all the photos. Mean Centering the dataset would remove this average mode.

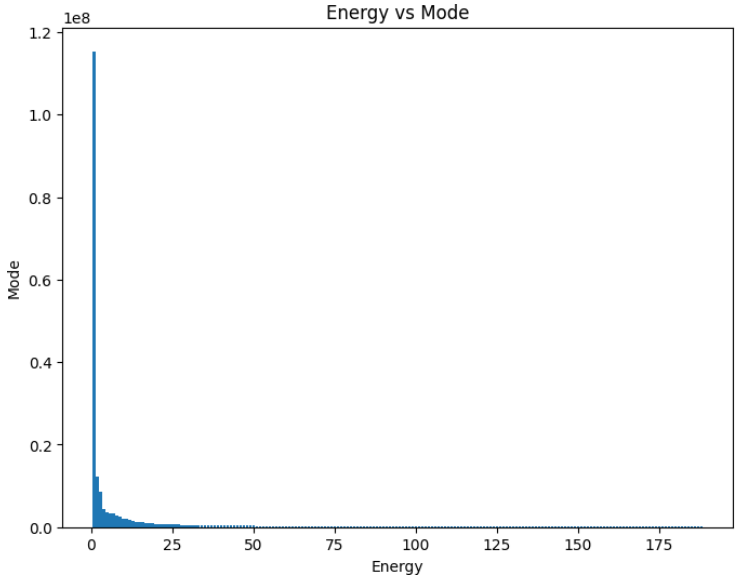

Figure 6: Energy vs Corresponding Mode Number for 188 Frames

2. Analysing Spatial Mode Matrix U

- Each column in the U matrix signifies how the flow is spread out in each mode. This matrix is basically the Basis Matrix of our simulation.
- Moving Left to Right in this matrix will show a decreasing in intensity as the magnitude of energy is decreasing.

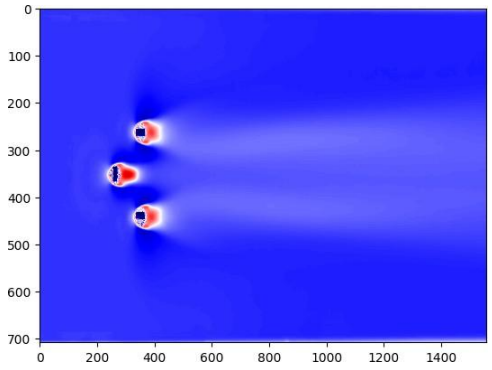

Figure 7: The Average Mode for 188 Frames

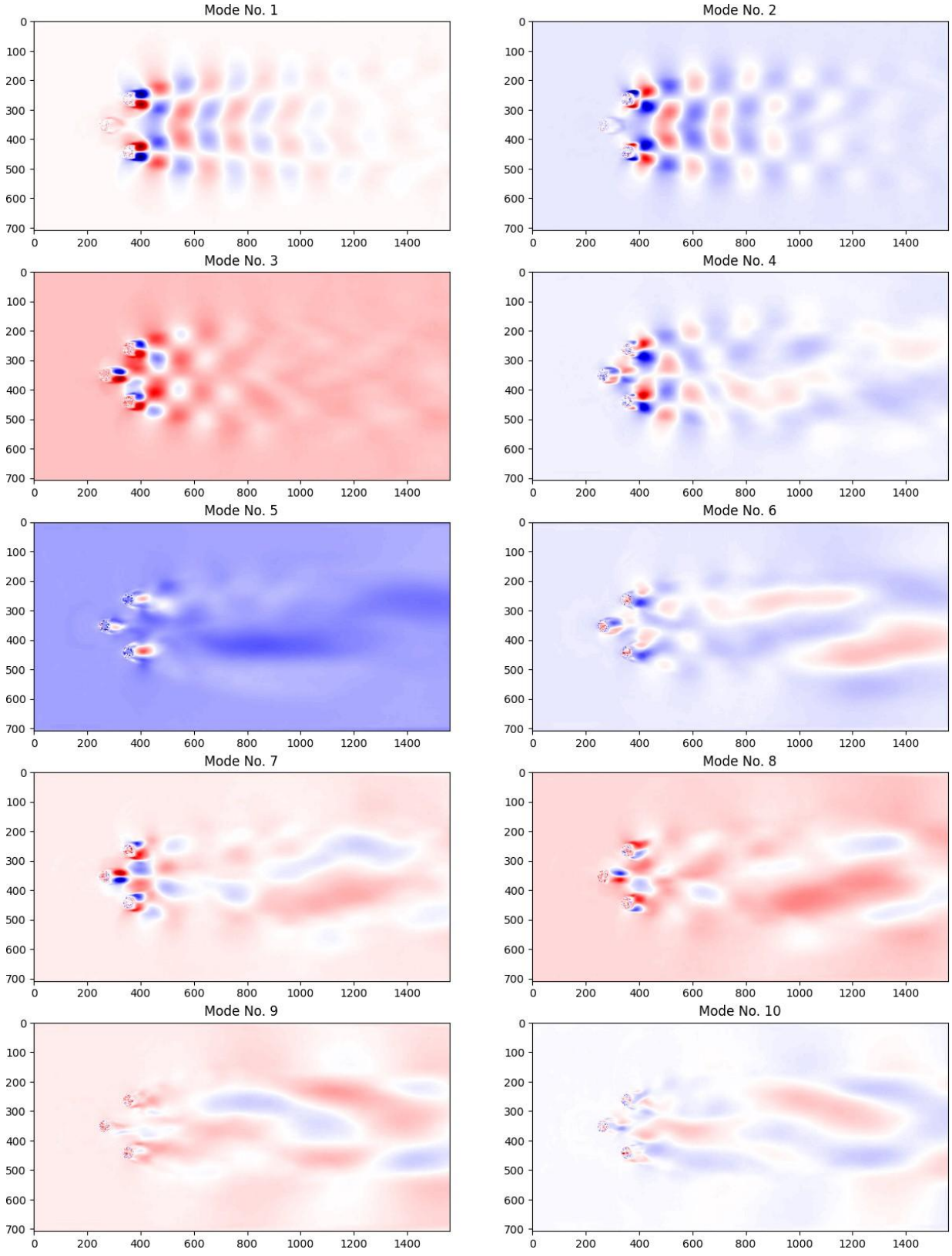

Figure 8: Energy vs Corresponding Mode Number for 188 Frames

3. Analysing Temporal Mode Matrix V_r^T

   - This matrix signifies the variation of intensity of each mode with time.
   - As seen in Figure 9 (a), Initially the mode is a periodic and then becomes periodic after some time. This is because the flow initailly begins from 0 to the constant velocity.

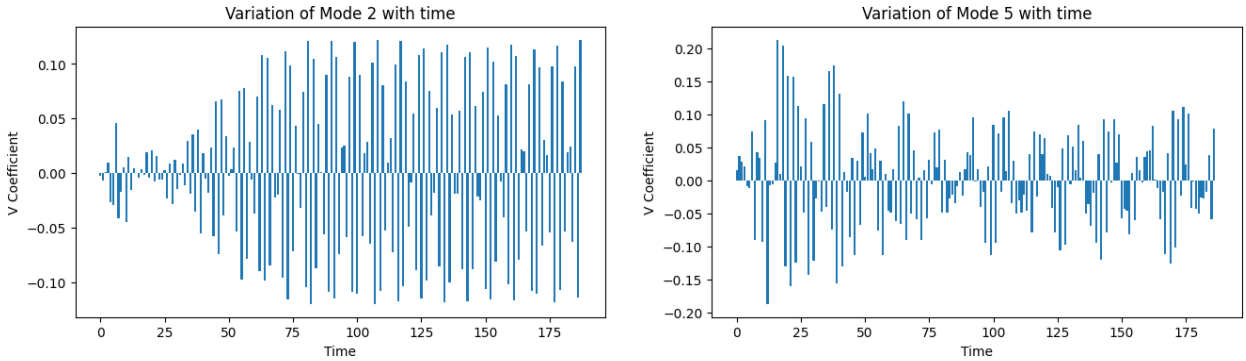

Figure 9: Variation of Intensity of Modes with Time

4 Noise

4.1 Adding Noise

We've incorporated two types of noise into our study: **Gaussian Noise** and **Salt-and-Pepper Noise**.

Gaussian Noise, named after Carl Friedrich Gauss, is a type of signal noise characterized by a probability density function (PDF) that aligns with the normal or Gaussian distribution. In essence, the noise values are Gaussian-distributed.

On the other hand, Salt-and-Pepper Noise, also known as impulse noise, is a type of noise often observed in digital images. It's typically caused by abrupt and sharp disturbances in the image signal.

Here's a detailed process of how we added these noises:

Gaussian Noise Addition Process: We first created a matrix of the same dimensions as our data image. Using a Gaussian probability distribution and a noise factor, we assigned a value to each pixel in this matrix. This Gaussian matrix was then added element-wise to our original image, resulting in what we refer to as the Noisy Gaussian Image.

Salt-and-Pepper Noise Addition Process: We first determined the number of pixels to which we needed to add noise, based on the desired noise magnitude. We then maximized half of these pixels and minimized the other half.This process resulted in the creation of Salt-and-Pepper noise in our image.

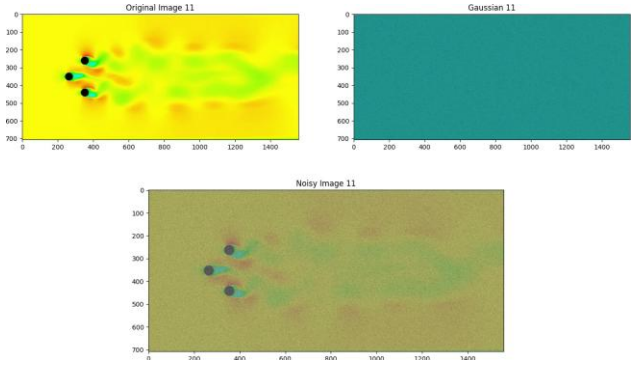

Figure 10: Gaussian Noise

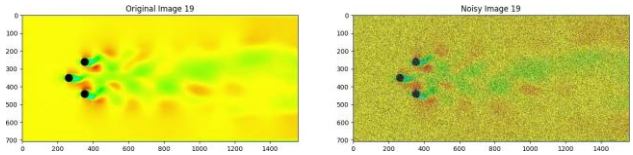

Figure 11: Salt and Pepper Noise

4.2 Effect on POD Modes

1. Effect of Gaussian Noise
    - There isn't a large effect on the energy in the modes due to gaussian noise.
    - Taking upto 60% cumulative variance is enough to regain our original dataset.
2. Effect of Salt Pepper Noise
    - Salt and pepper noise has a significant impact on the energy in the modes.
    - There is a sudden increase in energy across all modes results in significant noise when performing Reduced SVD, hence affecting the output.
3. Conclusion
    - From the images, it can be concluded that Gaussian noise best suits to POD while Salt Pepper cannot be suited as there is a large increase in energy of all modes due to Salt Pepper Noise.

5 Super-Resolving

We have used an Autoencoder as it is computationally less expensive than some of the other Deep Learning Algorithms such as RIDNet, ResNet, DNCNN etc.

Due to low computational resources, the image resolution was decreased and then padded by zeros to maintain a square shape.

Super-Resolution with Autoencoders: Enhancing Image Quality Super-resolution is a fascinating field in computer vision that aims to enhance the resolution and quality of images. Imagine taking a low-resolution image and transforming it into a high-resolution version, revealing finer details and improving visual clarity. Autoencoders, a type of neural network architecture, play a crucial role in achieving this goal.

1. **Autoencoders**:
    - An autoencoder is a neural network designed for unsupervised learning. It consists of an **encoder** and a **decoder**.
    - The encoder compresses the input data (such as an image) into a lower-dimensional representation (latent space).
    - The decoder reconstructs the original input from this compressed representation.
    - Autoencoders are often used for tasks like denoising, dimensionality reduction, and, in our case, super-resolution.
2. **Why Autoencoders for Super-Resolution?**:
    - Autoencoders are computationally less expensive compared to other deep learning algorithms like RIDNet, ResNet, or DNCNN.
    - Their simplicity makes them suitable for scenarios with limited computational resources.
    - By training an autoencoder on pairs of low-resolution and high-resolution images, we can learn to map low-res features to high-res features.
3. **The Process**:
    - **Data Preparation**:
        - We created a dataset of the same size as the original dataset but with either gaussian or salt pepper noise added to it.
        - Each image was resized to half its size keeping the aspect ratio constant for ease of computation. The images were cropped at edges to match the output matrix that the model would generate.
    - **The Model**:

Model Architecture		
Layer	Input Shape	No. of Parameters
Conv2D 1	(None, 352, 776, 32)	896
MaxPooling2D 1	(None, 176, 388, 32)	0
Conv2D 2	(None, 176, 388, 8)	2312
MaxPooling2D 2	(None, 88, 194, 8)	0
Conv2D 3	(None, 88, 194, 8)	584
MaxPooling2D 3	(None, 44, 97, 8)	0
Conv2D 4	(None, 44, 97, 8)	584
UpSampling2D 1	(None, 88, 194, 8)	0
Conv2D 5	(None, 88, 194, 8)	584
UpSampling2D 2	(None, 176, 388, 32)	0
Conv2D 6	(None, 176, 388, 32)	584
UpSampling2D 3	(None, 352, 776, 32)	0
Conv2D 7	(None, 352, 776, 3)	867

- **Training the Autoencoder**:
  - The autoencoder learns to encode the LR images into a compact representation.
  - During training, it minimizes the difference between the reconstructed HR images and the actual HR images.
- **Inference**:
  - Given a new LR image, we pass it through the trained encoder to obtain its latent representation.
  - The decoder then reconstructs the HR version of the image.
  - The output of our Model is shown in Figure 12
  - The Model can be compared using the metric Peak Signal-to-Noise Ratio (PSNR). The PSNR of the test images was 14.30892984386117 and for the denoised images was 19.702518474242034.

4. **Challenges and Trade-offs**:
   - While autoencoders are efficient, they may struggle with capturing intricate details.
   - Balancing computational cost and image quality is essential.
   - Researchers continually explore variations of autoencoders and hybrid approaches to achieve better super-resolution results.

In summary, autoencoders provide an elegant solution for super-resolving images, especially when computational resources are limited. By leveraging their ability to learn meaningful representations, we can enhance image quality and reveal hidden details.

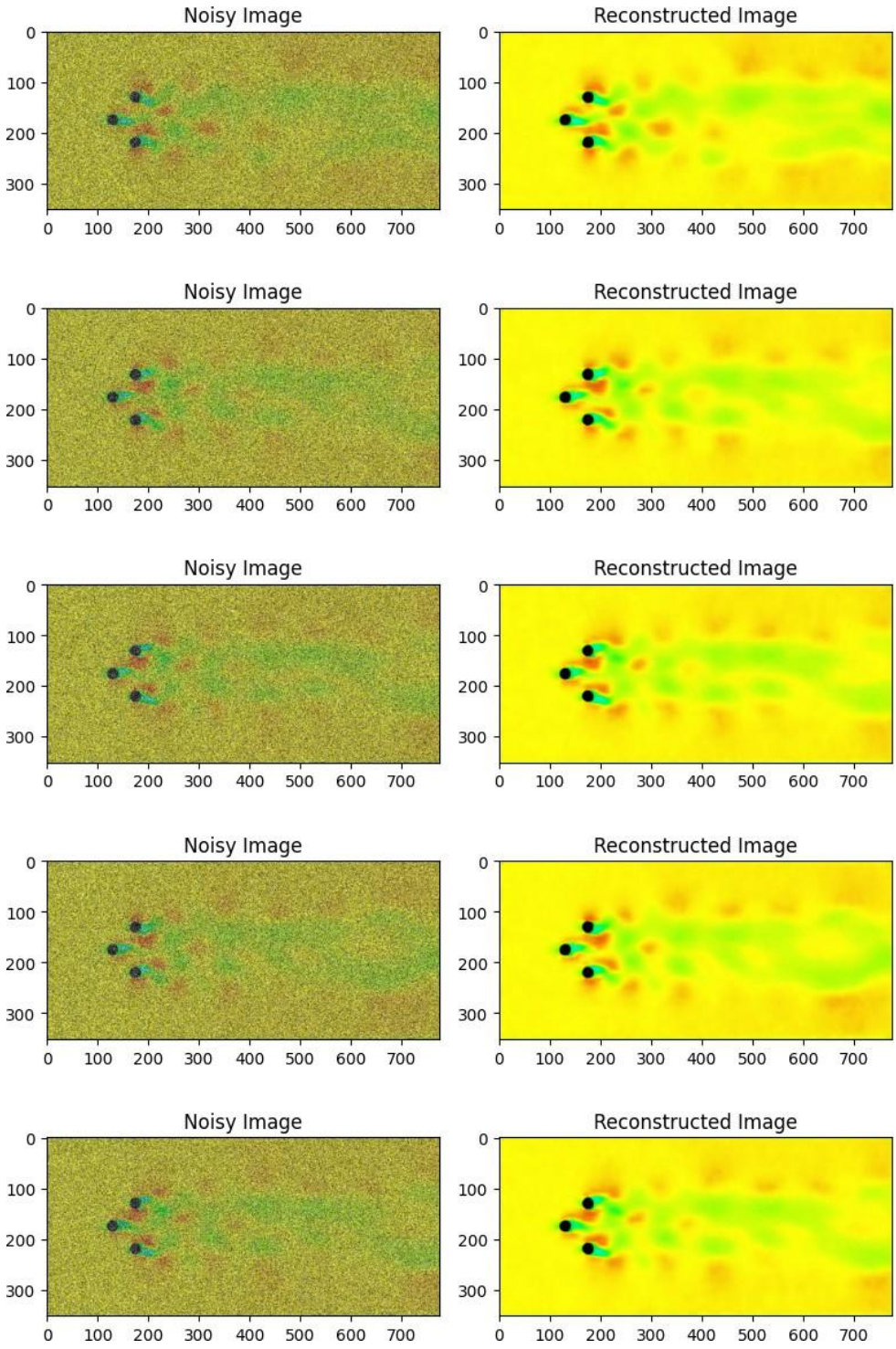

Figure 12: Output of our Model

Super-resolution is a critical task in image processing, where the goal is to enhance the resolution and quality of low-resolution images. In this report, we delve into the architecture of our super-resolution model, which employs an autoencoder-based approach. Our model aims to transform low-resolution images into high-resolution counterparts, revealing finer details and improving visual fidelity.

5.1 Model Architecture

Our model consists of several layers, each contributing to the overall reconstruction process. Let's break down the architecture:

1. **Input Layer (Conv2D)**:
    - The initial layer receives the low-resolution input image.
    - It employs a **2D convolutional operation** with **32 filters** and a **kernel size of 3x3**.
    - The output shape is **(None, 500, 500, 32)**.
    - Parameters: **896** (trainable weights).
2. **Max Pooling Layer (MaxPooling2D)**:
    - Subsampling layer that reduces the spatial dimensions.
    - Downsamples the feature maps by a factor of
    - Output shape: **(None, 250, 250, 32)**.
3. **Convolutional Layer (Conv2D)**:
    - Another 2D convolutional layer with **8 filters** and a **kernel size of 3x3**.
    - Output shape: **(None, 250, 250, 8)**.
    - Parameters: **2,312**.
4. **Max Pooling Layer (MaxPooling2D)**:
    - Further downsamples the feature maps.
    - Output shape: **(None, 125, 125, 8)**.
5. **Convolutional Layer (Conv2D)**:
    - Third 2D convolutional layer with **8 filters**.
    - Output shape: **(None, 125, 125, 8)**.
    - Parameters: **584**.
6. **Max Pooling Layer (MaxPooling2D)**:

- Reduces spatial dimensions again.
- Output shape: **(None, 63, 63, 8)**.

7. **Convolutional Layer (Conv2D)**:
   - Fourth 2D convolutional layer with **8 filters**.
   - Output shape: **(None, 63, 63, 8)**.
   - Parameters: **584**.
8. **Up-Sampling Layer (UpSampling2D)**:
   - Upsamples the feature maps back to higher resolution.
   - Output shape: **(None, 126, 126, 8)**.
9. **Convolutional Layer (Conv2D)**:
   - Fifth 2D convolutional layer with **8 filters**.
   - Output shape: **(None, 126, 126, 8)**.
   - Parameters: **584**.
10. **Up-Sampling Layer (UpSampling2D)**:
    - Further upsamples the feature maps.
    - Output shape: **(None, 252, 252, 8)**.
11. **Convolutional Layer (Conv2D)**:
    - Sixth 2D convolutional layer with **32 filters**.
    - Output shape: **(None, 250, 250, 32)**.
    - Parameters: **2,336**.
12. **Up-Sampling Layer (UpSampling2D)**:
    - Upsamples to the original image resolution.
    - Output shape: **(None, 500, 500, 32)**.
13. **Output Layer (Conv2D)**:
    - Final 2D convolutional layer with **3 filters** (for RGB channels).
    - Reconstructs the high-resolution image.
    - Output shape: **(None, 500, 500, 3)**.
    - Parameters: **867**.

5.2 Conclusion

Our autoencoder-based super-resolution model learns to map low-resolution features to high-resolution features. Despite its simplicity, it provides a computationally efficient solution for enhancing image quality. Researchers continue to explore variations and hybrid approaches to strike a balance between computational cost and image fidelity.

6 Reference Material

[1] Fernando Zigunov. (May 12, 2021). Understanding POD: the Proper Orthogonal Decomposition [Video file]. YouTube. https://www.youtube.com/watch?v=axfUYYNd-4Y&ab_channel=FernandoZiguno

[2] Roshan Joe Vincent. (Jul 29, 2021). Singular Value Decomposition (SVD) — Working Example. Medium. https://medium.com/intuition/singular-value-decomposition-svd-working-example-c2

[3]https://www.rivieramm.com/news-content-hub/news-content-hub/how-ai-tackles-shipping-chaos-and-p

[4]https://inria.hal.science/hal-01771298/document

[5]https://youtu.be/8e3OT2K99Kw?si=ccVivLaZ404sPSxc

[6]https://en.wikipedia.org/wiki/Gaussiannoise

[7] https://www.geeksforgeeks.org/add-a-salt-and-pepper-noise-to-an-image-with-python/

[8] Danso-Amoako, E.; Scholz, M.; Kalimeris, N.; Yang, Q.; Shao, J. Predicting dam failure risk for sustainable

[9] flood retention basins: A generic case study for the wider greater manchester area. Comput. Environ. Urban Syst. 2012, 36, 423–433. [CrossRef]

[10] Xie, K.; Ozbay, K.; Zhu, Y.; Yang, H. Evacuation zone modeling under climate change: A data-driven method.

[11]J. Infrastruct. Syst. 2017, 23, 04017013. [CrossRef] Pitt, M. Learning Lessons from the 2007 Floods; Cabinet Office: London, UK, 2008.

[12]Lohani, A.K.; Goel, N.; Bhatia, K. Improving real time flood forecasting using fuzzy inference system.

[13]J. Hydrol. 2014, 509, 25–41. [CrossRef] Mosavi, A.; Bathla, Y.; Varkonyi-Koczy, A. Predicting the Future Using Web Knowledge

[14]Art Survey. In Recent Advances in Technology Research and Education; Springer: Cham, Switzerland, 2017; pp. 341–349.

[15]Engineering Research Express 2023-12-01 — Journal article DOI: 10.1088/2631-8695/ad0bdc CONTRIBUTORS: Rushil Samir Patel; Harshal D Akolekar