MEL2040 Project
Data Driven Analysis of Fluid Flows

Yuvraj Saran & *Jyoti*

April 15, 2024

Introduction

The project aims to study the integartion of machine learning and fluid mechanics i.e, use of machine learning in the domain of fluid mechanics . it starts with analysing the research papers based on the novalty of ML algorithms in fluid mechanics and follows up with some new ideas where we can use ML algorithms in fluid mechanics domain.it mainly focuses on the application on POD / PCA that breaks down complex system into simpler parts. PCA uses eigen values and eigen vectors to simplify data dimensions whereas POD uses singular value decomposition (SVD) so that spatial aspects can be understood better. we have done tasks like image generation , plot representaions using POD , analysing how noise is affecting our model analysing data by adding different magnitudes of noise and then performing POD on the noisy images , and then removing the noise using PCA and applied POD on denoised images and generated graphs for the top energy modes.

1 Review of Machine Learning in Fluids

Litreture review of how machine learning is used in fluid mechanics domain) in fluid mechanics , we have a vast and increasing amount of data from simulation and experiments. as we already know most of the fluid mechanics task involve reduction , modeling, control,closure. machine learning is basically building models from data using optimizaion.we need models that are interpretable and generalizable so that we can understand and communicate and works with models .

1.1 Use of machine learning algorithms for CFD in sense of DNS

Navier-stokes equation is have number of scales in time and space so it is a multiscale equation and it is expensive to get solved by the CFD solver and also it takes very long time to simulate even the simplified version of flows that are not even complex like some we see in our real world so machine learning algorithms can be used to solve the CFD problems in less time and reduced cost .

ML algorithms can also be used to accelerate the computation of poisson equation or a laplace equation (that is also expensive to be solved by a CFD solver) fully convolutional neural networks can be used to solve a poisson equation by breaking the poisson problem into a homogenous poisson problem and four inhomogeneous laplace subproblems this also made the computation cheaper .

1.2 Use of machine learning algorithms for turbulence closure modeling

Turbulence modeling is an important aspect of fluid dynamics and it is hard to calculate and have different forms . Reynolds-averaged navier-stokes equation (RANS) is the reduced form of general navier-stokes equation and is one of the most popular turbulence closure modeling frameworks; in RANS the steady state solution is decoupled from the time varying fluctuations in the system and RANS equation consist of a reynolds stress term which depend on the temporal fluctuations so the use of machine learning is to model the reynolds stresses which is a important tern to get turbulence closure model . One another approach for the turbulence closure modeling is large eddy simulations (LES), Most LES studies look at the largest sizes of turbulent flows while RANS models do not . so we can use machine learning algorithms for improving turbulence models

1.3 Use of machine learning algorithms for reduced order models (ROMs)

In fluid mechanics machine learning is also used to make reduced order models (ROMs) these reduced order models are used to show how the patterns change over time provide us details and the accurate picture of the fluid , as the computational fluid dynamics methods are expensive and takes a long time to be solved so reduced order model replace them ,we can make reduced order model by finding a set of coordinates that describe the amplitude of different flow structures and differential equation models that shows how the amplitude changes over time and adding machine learning algorithms to this can make process of turbulent modeling faster and more accurate . There are two methods that can be used in data driven modeling and those are Principal component analysis (PCA) Singular value decomposition (SVD) These two methods reduces down the number of variables

1.4 Machine learning algorithm for flow control

Machine learning algorithms can also be used for flow control in fluid mechanics , neural networks are used for turbulence flow control initially the skin friction drag of a turbulent boundary layer was reduced using local wall - normal blowing and suction based on few skin friction sensors. The single-layer neural networks optimization is done for skin-friction drag reduction without any assumptions used of actuation commands . Reinforcement learning algorithms can also be used for flow control by manipulating the parameters to achieve desired objectives

2 Any New Ideas (how machine learning can be applied in fluid mechanics domain)

2.1 Use of classification in fluid mechanics

classification which is a supervised learning method which studies the behavior of data and classifies them into categories, in fluid mechanics classification can be used to distinguish between various canonical behaviors and dynamic regimes . Previously classification of wake topology behind a pitching airfoil from local vorticity measurement using a neural network was done and also KNN was used to detect exotic wakes .So neural networks can be combined with dynamic systems models to detect flow disturbances and estimate their parameters .

2.2 Flow Control Optimization with Reinforcement Learning

to optimize flow control strategies we can use reinforcement learning method s. Flow control involves manipulates fluid flows to so that it can get its required outocomes like reducing drag and enhancing mixing . Reinforcement learning algorithms can learn optimal control policies by interacting with simulated or experimental flow systems. These learned policies can then be applied to real-world fluid systems for efficient flow control.

2.3 Predictive Maintenance for Fluid Systems

we can use machine learning for predictive maintenance for fluid system. ML algorithms have the ability to analyze data and tools used in fluids like,valves, and pipelines can be used for predicting potential failures. we can reduce downtime, improve safety, and optimize maintenance schedules if we are able to predict maintenance in advance

2.4 optimized design exploration using generative models

generative machine learning models can be used for optimizing and exploring fluid system design. generative adversarial networks(GANs) or variational autoencoders(VAEs) are some generative models that can help to generate novel designs.these models can help us in exploring a large range of designs possibilities like pressure drop , heat transfer etc

3 Proper Orthogonal Decomposition

3.1 Image generation

the given task for image generation we are provided a video in the question discription itself we have taken that video as input , the duration of the given video was 31 sec in which the frame rate was 24 fps and we have extracted a total of 751 frames including the initial frame and we have saved every 15th frame of the video . we have saved a total of 51 frames. Refer Figure 1

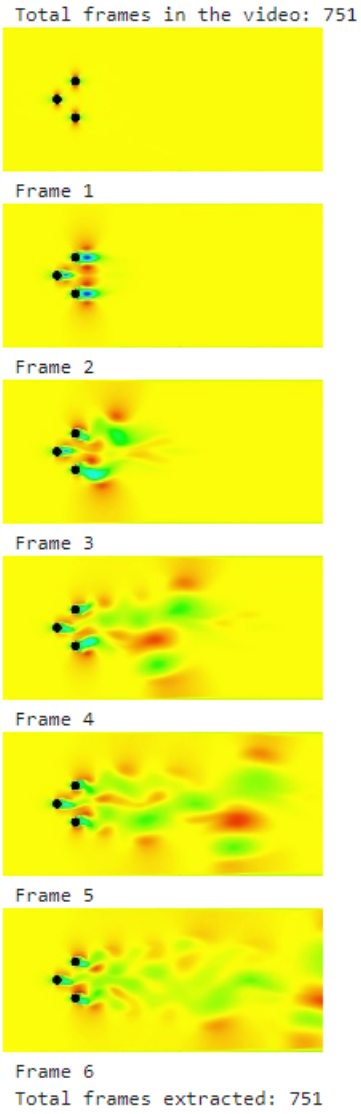


Figure 1: Snaps of the images from the provided video

3.2 Execute POD

in the given task we have to tell the mathamatical formulations used so in this section we have calculated the covariance matrix using the formula and then we have performed Eigen value decomposition on the covariance matrix to find the eigen-value and eigen-function.and following to that we have eigenvectors corresponding to largest eigen values are selected . The formulae used for the task are - formula for covariance matrix-

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

```python
def compute_covariance_matrix(snapshot_matrix):
    # Compute covariance matrix in batches to reduce memory usage
    batch_size = 100  # Adjust batch size as needed
    num_samples = snapshot_matrix.shape[0]
    covariance_matrix = np.zeros((snapshot_matrix.shape[1], snapshot_matrix.shape[1]))
    for i in range(0, num_samples, batch_size):
        batch = snapshot_matrix[i:i+batch_size, :]
        covariance_matrix += np.dot(batch.T, batch)
    covariance_matrix /= (num_samples - 1)
    return covariance_matrix

# Compute covariance matrix
covariance_matrix = compute_covariance_matrix(snapshot_matrix)

def eigen_decomposition(covariance_matrix):
    eigenvalues, eigenvectors = np.linalg.eigh(covariance_matrix)
    return eigenvalues[::-1], eigenvectors[:, ::-1]  # Reverse order to get descending eigenvalues

# Perform eigenvalue decomposition
eigenvalues, eigenvectors = eigen_decomposition(covariance_matrix)
```

Figure 2: Code used for calculating covariance matrix and applying eigen-decomposition on the snapshots

$\bar{x}$ represents the mean of all snapshots. x_i is the i-th snapshot. N is the total number of snapshots. Eigenvalue decomposition is performed using the equation:

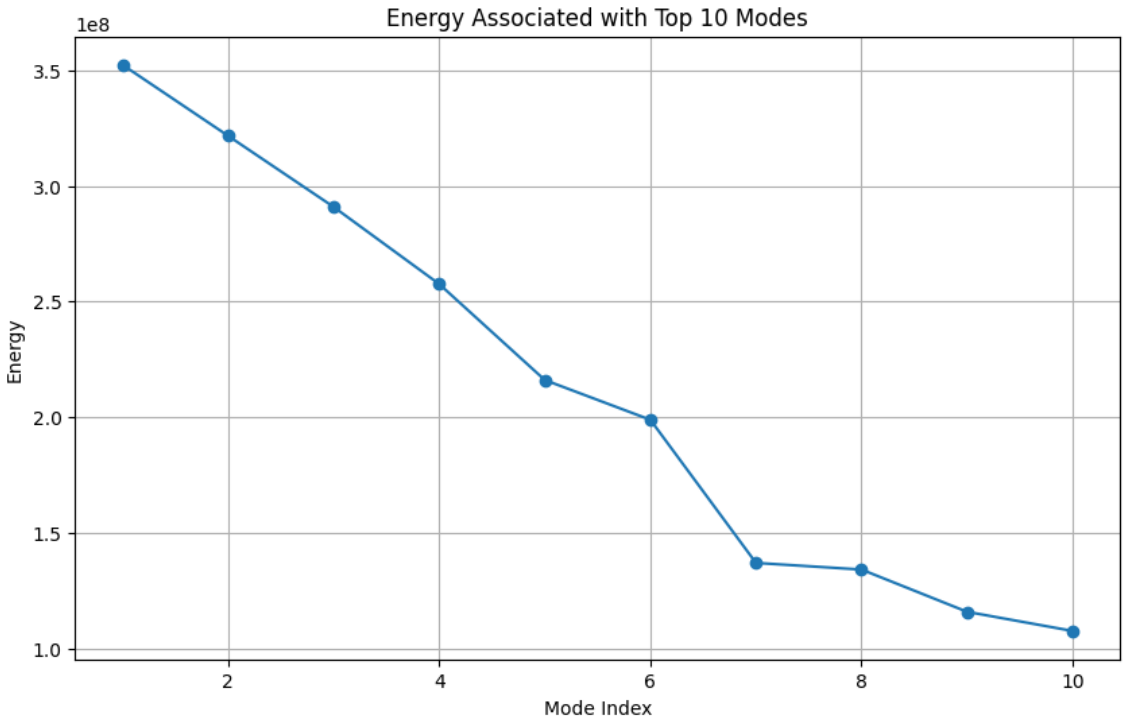
$$\mathbf{Cv} = \lambda \mathbf{v}$$

code used for computing the covariance matrix is given below

As we have studied in research papers while finding the use of Machine learning algorithm in fluid mechanics that ML algorithms can be used to find spatial patterns so here in the code used for this task it has selected most significant spatial patterns in the data and additionally code calculates and prints the energy of each mode ,so we can say that the code understands the underlying spatial and temporal structures in the dataset using POD analysis . finding the modes can help us simplifying the complex data and to understand the system's dynamics

3.3 Analyse POD Modes

To visualize energy distribution among the modes we have plotted the graph. Refer Figure 3. The graph shows trend of decreasing energy as we move from higher modes to lower modes clearly showing that the dominant flow features are captured by the first few modes.




```
Top 10 energy states (modes):
Mode 1: Energy = 352374824.5365
Mode 2: Energy = 321664859.6388
Mode 3: Energy = 290899214.8431
Mode 4: Energy = 257543992.9928
Mode 5: Energy = 216054397.3808
Mode 6: Energy = 198854147.1853
Mode 7: Energy = 136969399.7115
Mode 8: Energy = 134091520.7423
Mode 9: Energy = 115824563.7397
Mode 10: Energy = 107536430.1055
```

Figure 3: Energy of Top 10 Modes of True images

These modes represent the most energetically significant flow structures in the system. The reason these top modes are essential is that they address the relevant flow characteristics while reducing the dimensionality of the data. They are the main flow drivers, which implies that the majority of pertinent information regarding the narrative stream behavior is contained in these top rated modes. Engineers and research frequently rely on dominant modes ranked in MMs to identify the leading flow attributes and predict flow evolution.

4 Noise!

analysing how noise is effecting our model studies. We have applied 3 types of noise : Gaussian Noise, Salt and Pepper Noise, and Speckle Noise

4.1 Adding noise

There are three types of artificial noises that we have explored by adding in the frames that we have extracted.we have added every noise with the magnitude (20, 40, 60, 80)percent in our originally extracted image

Gaussian noise-It is a statistical noise with a probablity density function equal to the normal distribution. Gaussian noise has a uniform distribution throughout the signal.

Adding 20 percent Noise

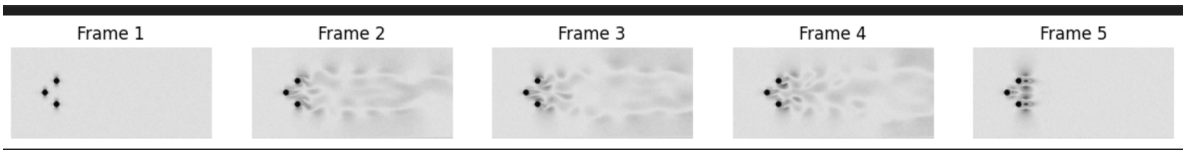


Figure 4: the figure shows the frames with 20 percent of gaussian noise

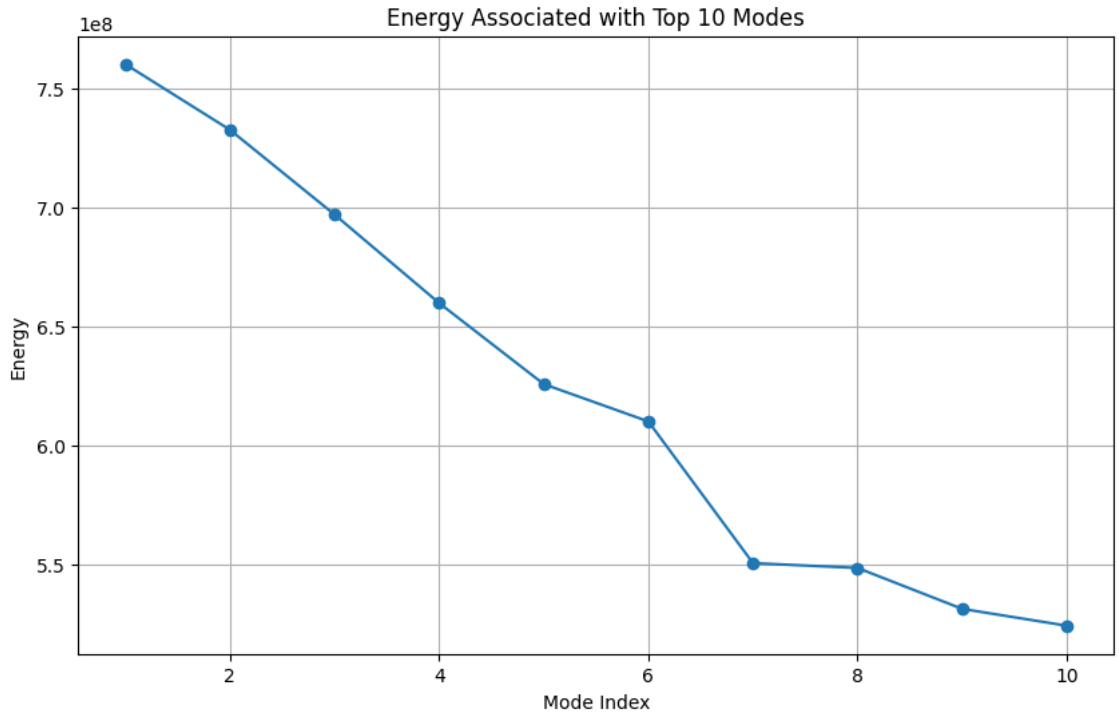


Figure 5: this figure shows top 10 modes modes for 20 percent of gaussian noise

Top 10 energy states (modes):
Mode 1: Energy = 760459411.8541
Mode 2: Energy = 732858199.8516
Mode 3: Energy = 697220014.0859
Mode 4: Energy = 660001184.4510
Mode 5: Energy = 625914328.4233
Mode 6: Energy = 610211675.7642
Mode 7: Energy = 550641172.0171
Mode 8: Energy = 548587915.4974
Mode 9: Energy = 531393874.8667

Mode 10: Energy = 524314757.6261

Adding 40 percent Noise

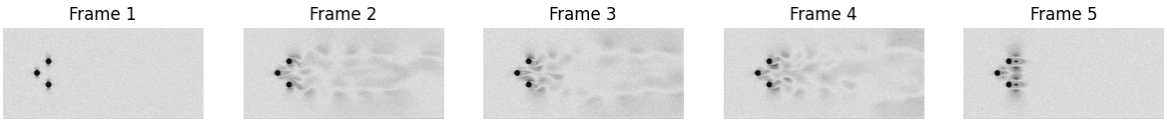


Figure 6: the figure shows the frames with 40 percent of gaussian noise

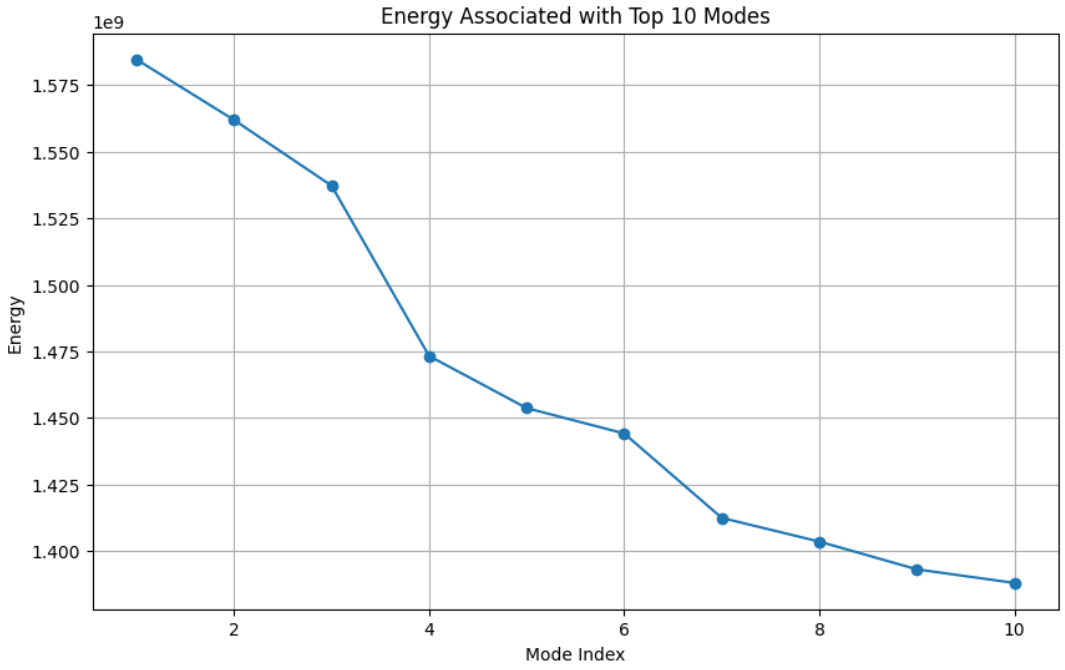


Figure 7: this figure shows top 10 modes modes for 40 percent of gaussian noise

Top 10 energy states (modes):
Mode 1: Energy = 1584637787.9769
Mode 2: Energy = 1561912650.0631
Mode 3: Energy = 1537236725.0471
Mode 4: Energy = 1473285251.4626
Mode 5: Energy = 1453715616.4721
Mode 6: Energy = 1444182329.2549
Mode 7: Energy = 1412488400.4551
Mode 8: Energy = 1403585713.5426
Mode 9: Energy = 1393163789.5016
Mode 10: Energy = 1388059458.5900

Adding 60 percent Noise

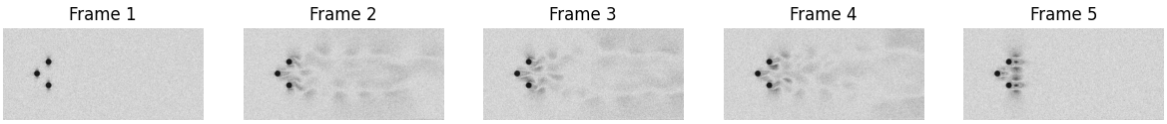


Figure 8: the figure shows the frames with 60 percent of gaussian noise

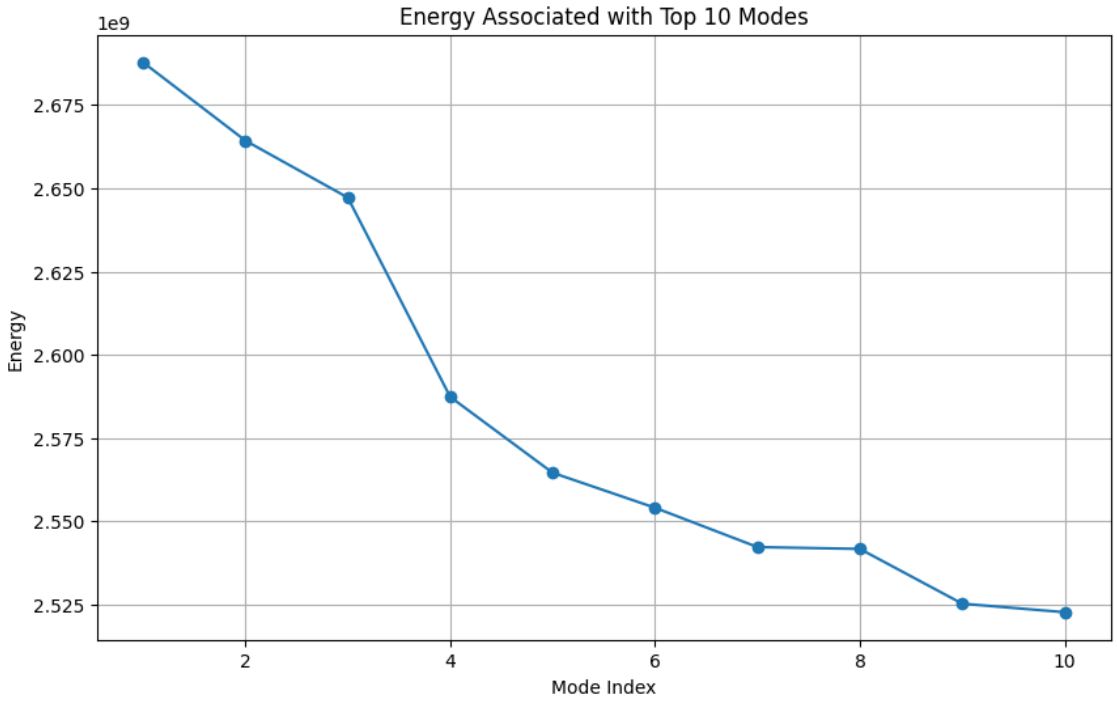


Figure 9: this figure shows top 10 modes modes for 60 percent of gaussian noise

Top 10 energy states (modes):
Mode 1: Energy = 2687790478.3218
Mode 2: Energy = 2664279350.7385
Mode 3: Energy = 2647175076.2059
Mode 4: Energy = 2587361848.3065
Mode 5: Energy = 2564590882.9957
Mode 6: Energy = 2554092562.4268
Mode 7: Energy = 2542295478.5582
Mode 8: Energy = 2541738823.6396
Mode 9: Energy = 2525304220.9987
Mode 10: Energy = 2522742105.3515

Adding 80 percent Noise

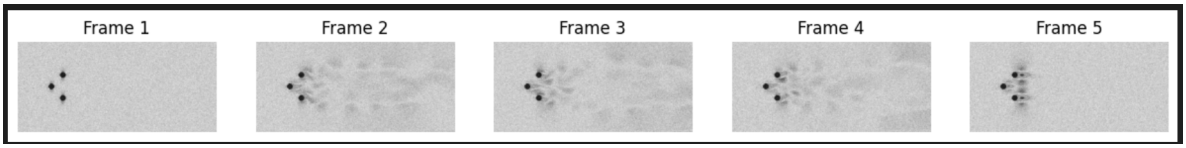


Figure 10: the figure shows the frames with 80 percent of Gaussian noise

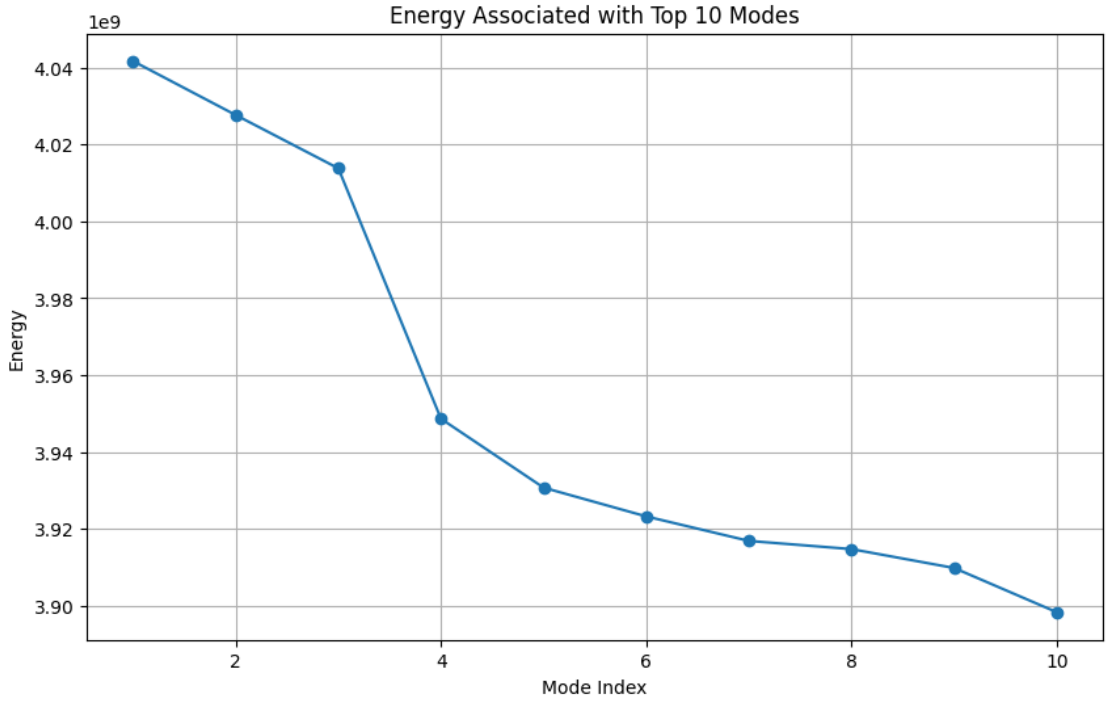


Figure 11: this figure shows top 10 modes modes for 80 percent of gaussian noise

Top 10 energy states (modes):
Mode 1: Energy = 4041676367.4666
Mode 2: Energy = 4027658078.3361
Mode 3: Energy = 4013790214.7325
Mode 4: Energy = 3948663756.9848
Mode 5: Energy = 3930746034.2020
Mode 6: Energy = 3923268932.4906
Mode 7: Energy = 3916898060.8766
Mode 8: Energy = 3914768261.9951
Mode 9: Energy = 3909853166.2539
Mode 10: Energy = 3898352633.2271

Salt and pepper noise - this noise produces random black and white pixels which generate localized disruptions. As a result, it is expected not to be able to observe small-scale flow structures, and artifacts were created in the modes. This type of noise is commonly seen in photographs this is caused by error in data transfer

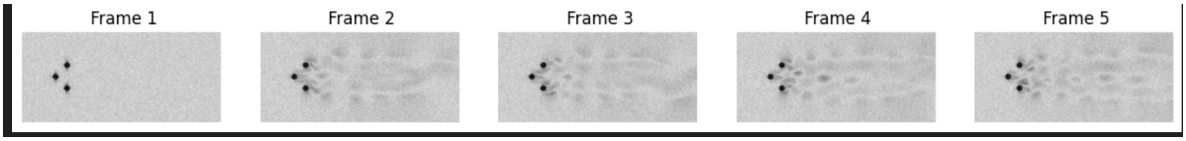


Figure 12: the figure shows the frames with 20 percent of salt and pepper noise

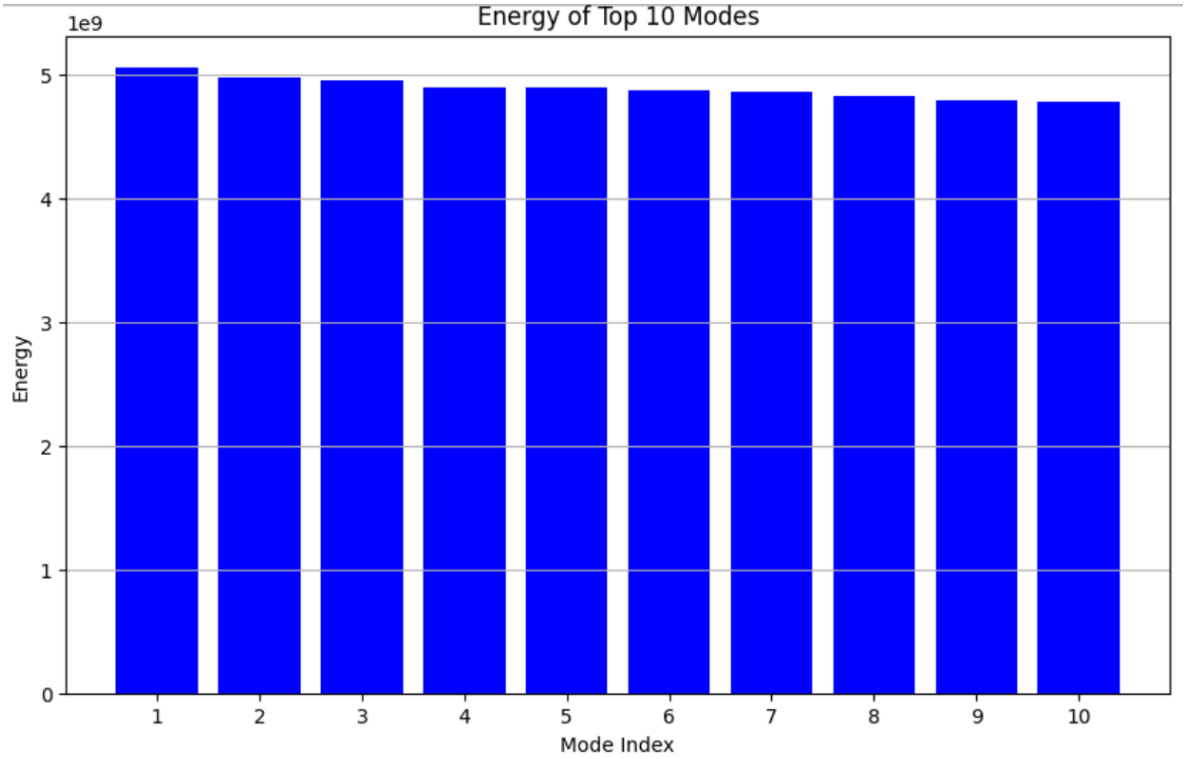


Figure 13: the fig shows the top 10 energy modes for 20 percent of salt and pepper noise

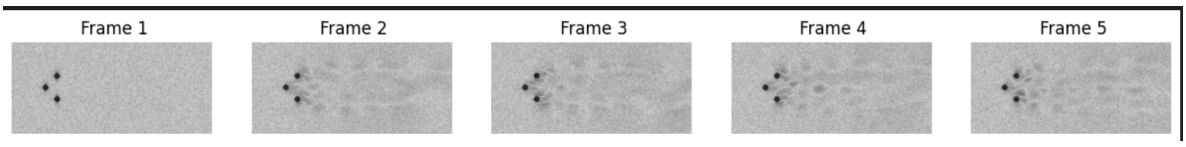


Figure 14: the figure shows the frames with 80 percent of salt and pepper noise

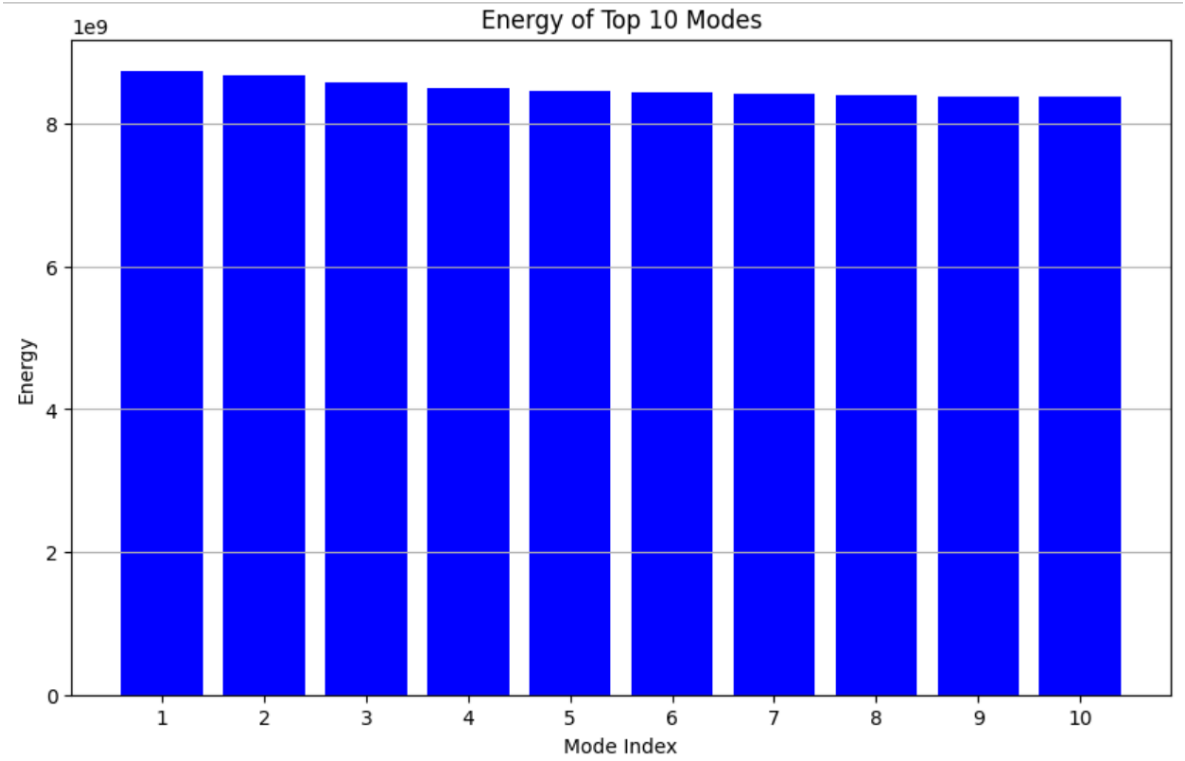


Figure 15: the fig shows the top 10 energy modes for 80 percent of salt and pepper noise

Speckle noise - This noise produced grain-like patterns across the picture which appeared like interference or texture. Although speckle noise was not likely to influence polarization features, it generates disturbances which make mode interpretation challenging.

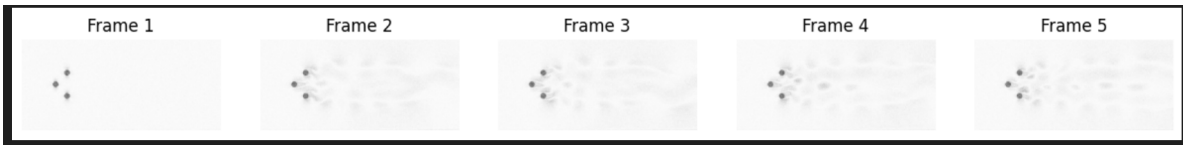


Figure 16: the figure shows the frames with 20 percent of the speckle noise

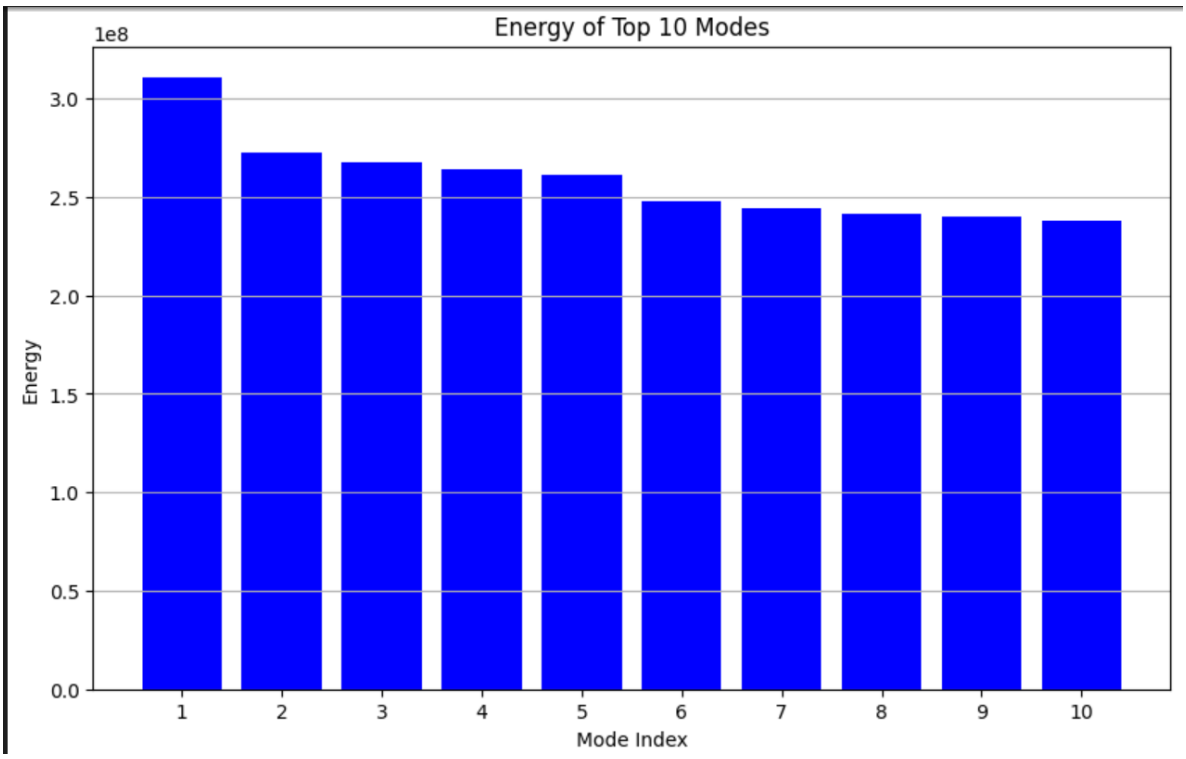


Figure 17: this figure shows the top 10 energy modes for the 20 pecent of speckle noise

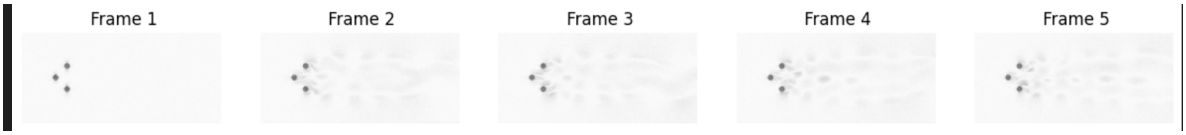


Figure 18: the figure shows the frames with 80 percent of speckle noise

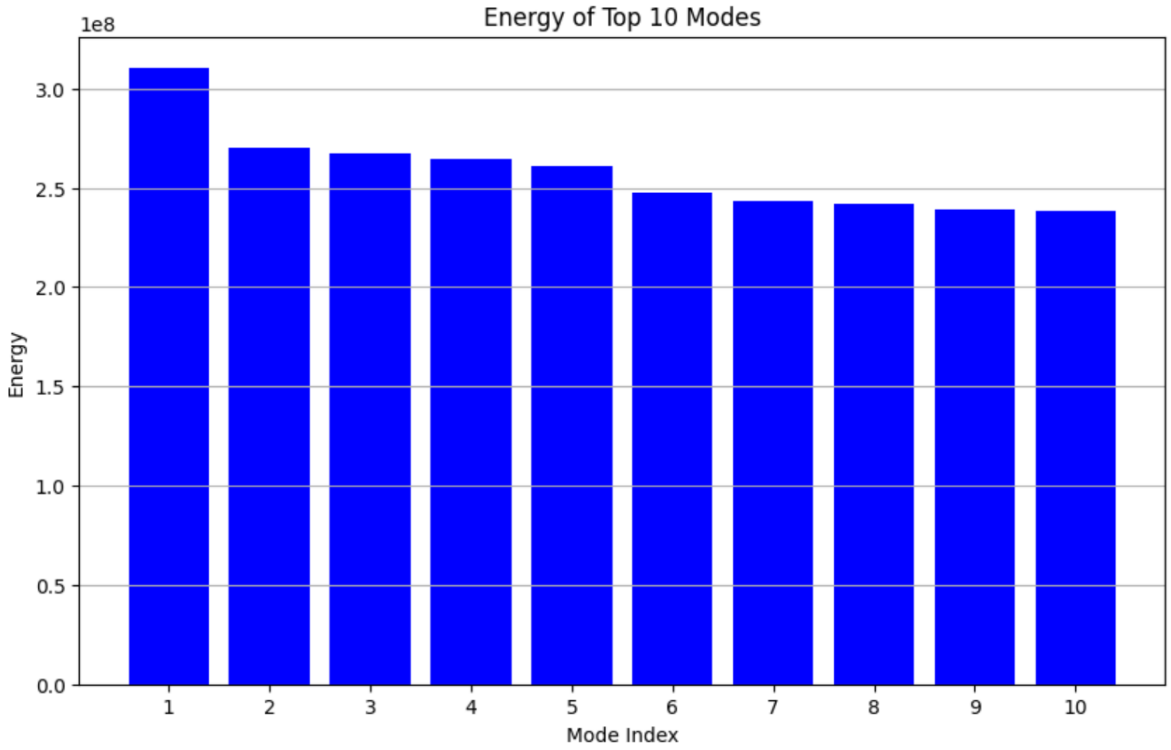


Figure 19: the fig shows the top 10 energy modes for 80 percent of speckle noise

4.2 Effect on POD modes

Yes, different noise types have different type of response in terms of how the energy of the modes changes the effect of magnitude of noise on modal energy. If we increase the magnitude of noise then the energy for higher order modes will increase and for the dominant modes there will be a decrease in energy .The reason behind it is that noise with strong magnitude introduces more variations across the data.

The noise which is best suited to be used for POD for artificial analysis is Gaussian noise because it is uniformly distributed and it has a control over intensity and it also have the ability to maintain a constant randomness in the dataset.

5 Super-Resolving

We have used PCA to remove the noise form the images.
Here is the code for removing noise using PCA

```
# Function to extract features using PCA
def extract_features(images, num_components):
    """
    Extract features from images using Principal Component Analysis (PCA).
    :param images: Input images.
    :param num_components: Number of principal components to keep.
    :return: PCA object and extracted features.
    """
    # Flatten the images
    flattened_images = images.reshape(len(images), -1)

    # Perform PCA
    pca = PCA(n_components=num_components)
```

```
    pca.fit(flattened_images)

    # Transform the images into the PCA space
    features = pca.transform(flattened_images)

    return pca, features
```

The denoised images we get after super resolving are :

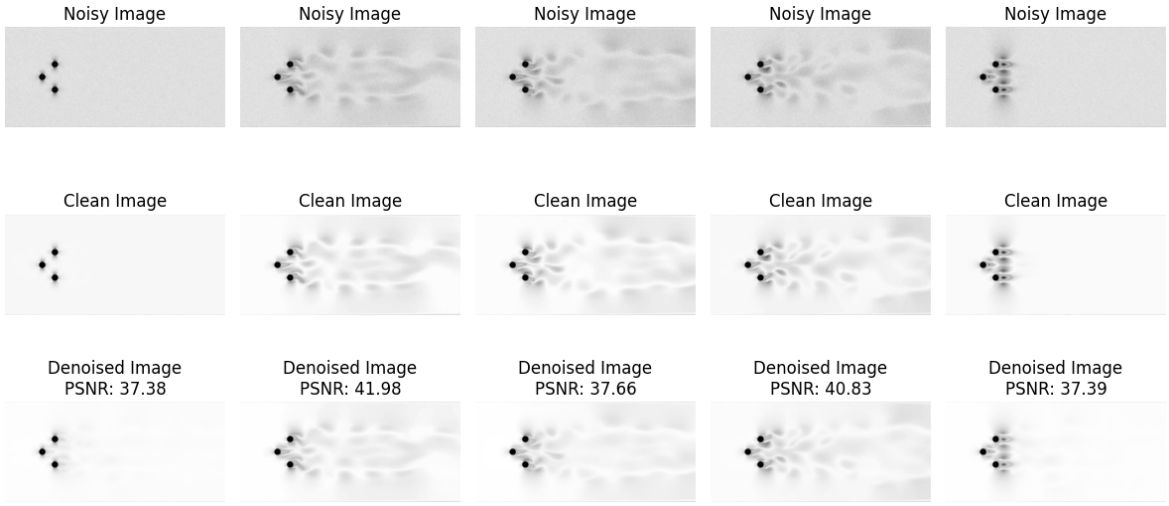


Figure 20: 20 percent of gaussian noise

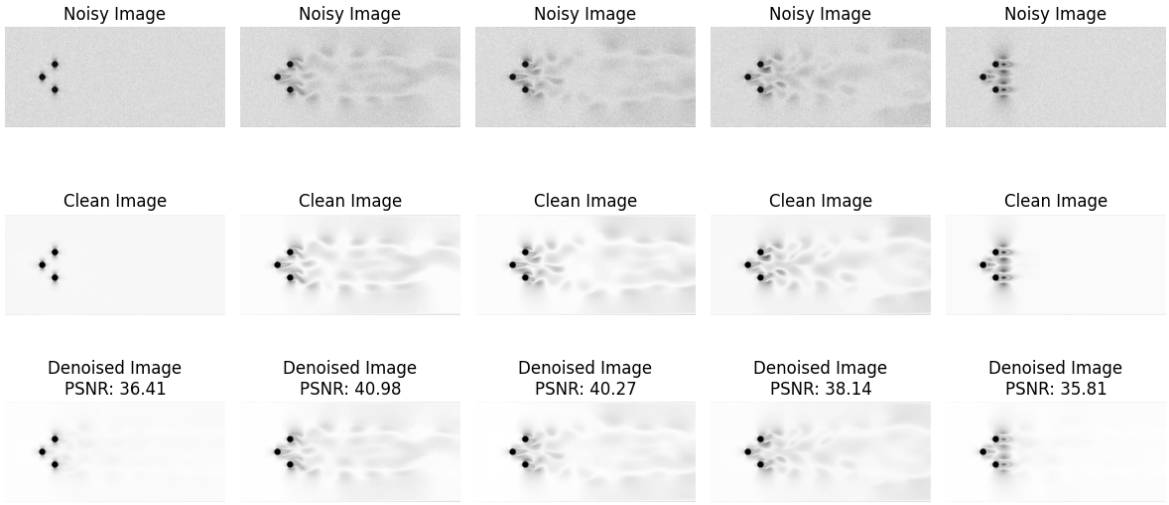


Figure 21: 40 percent of gaussian noise

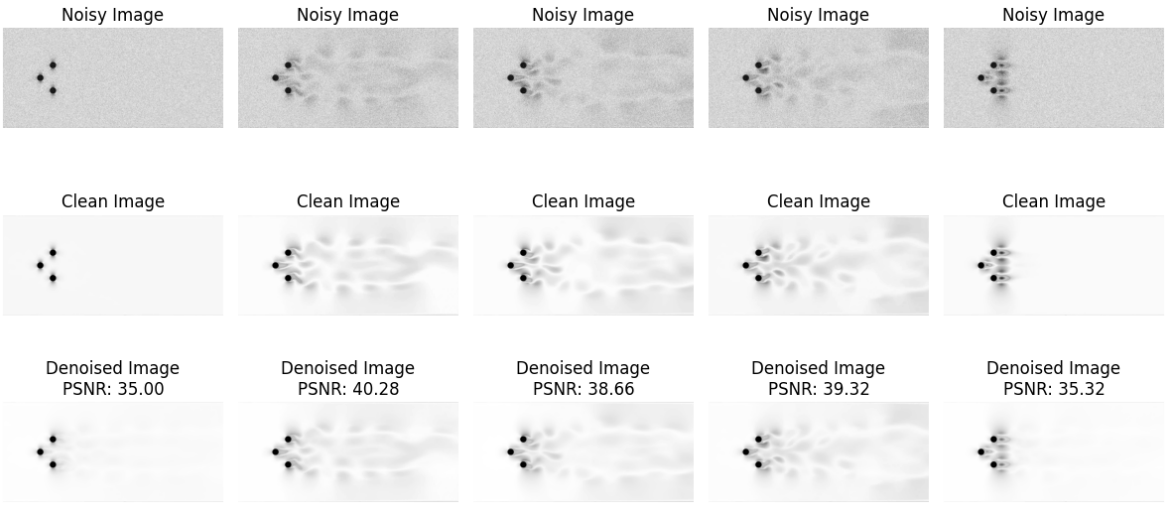


Figure 22: 60 percent of gaussian noise

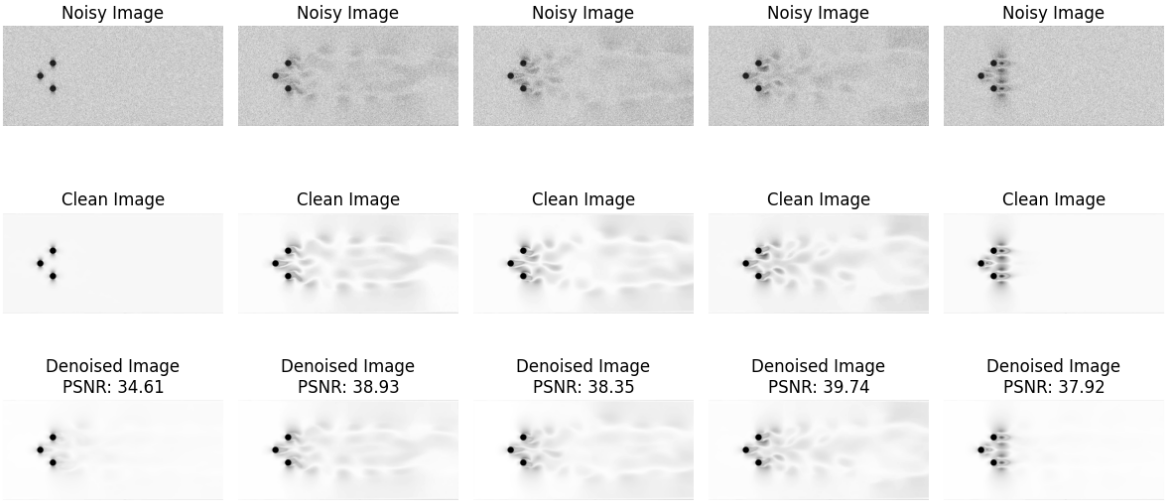


Figure 23: 80 percent of gaussian noise

And the graphs for top 10 energy modes for respective noise are:

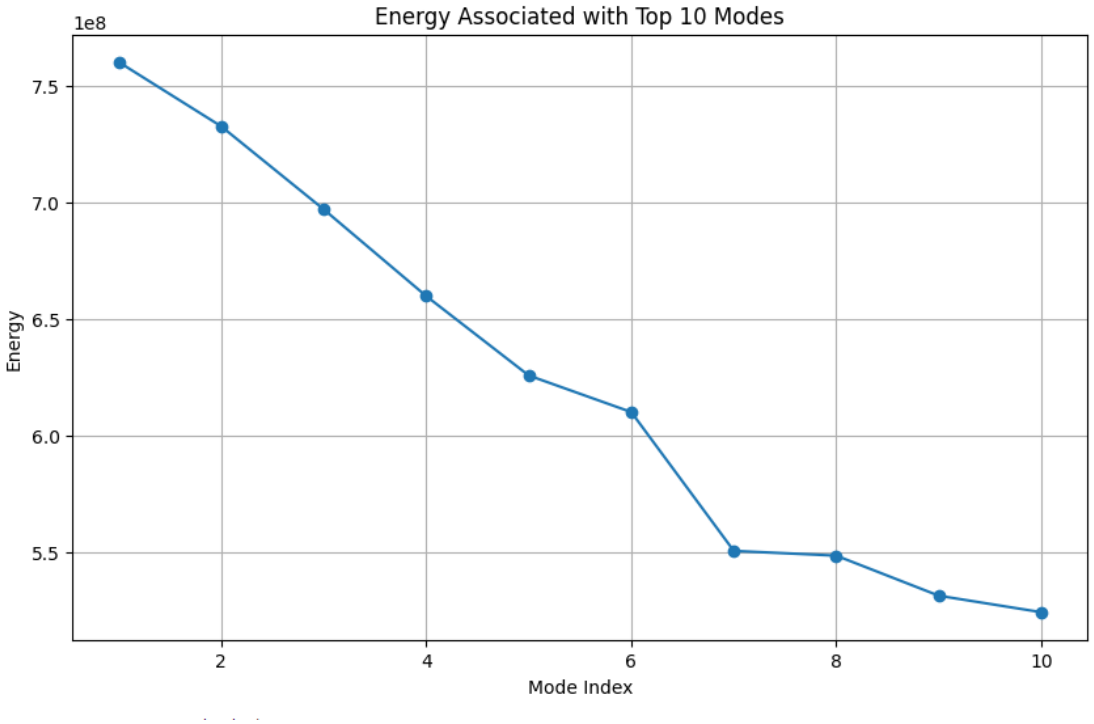


Figure 24: Top 10 energy modes for 20 percent of gaussian noise

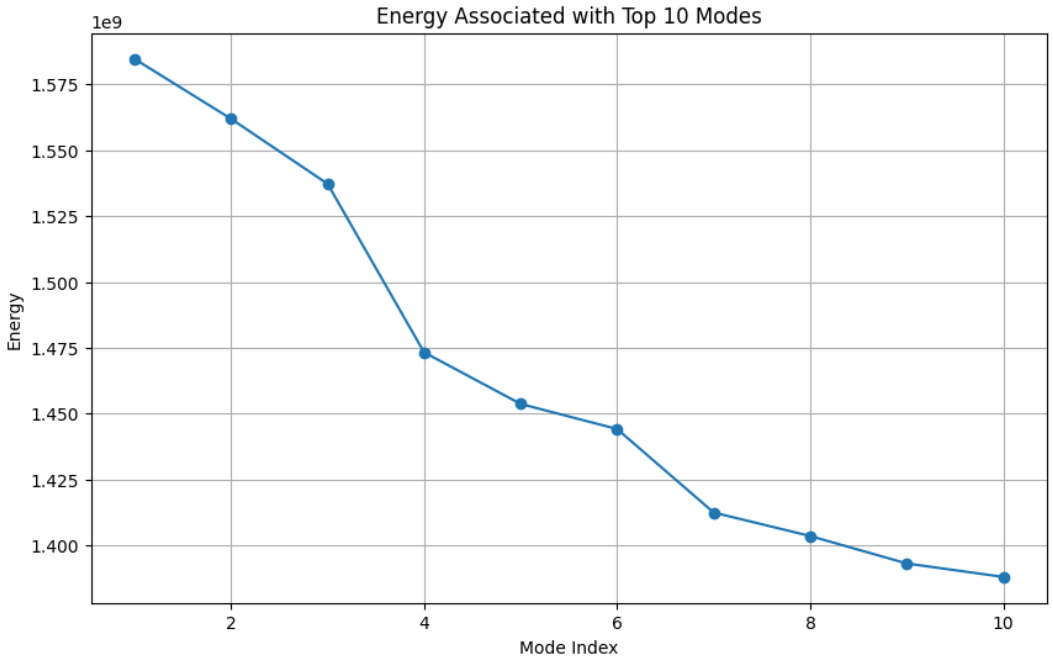


Figure 25: Top 10 energy modes for 40 percent of gaussian noise

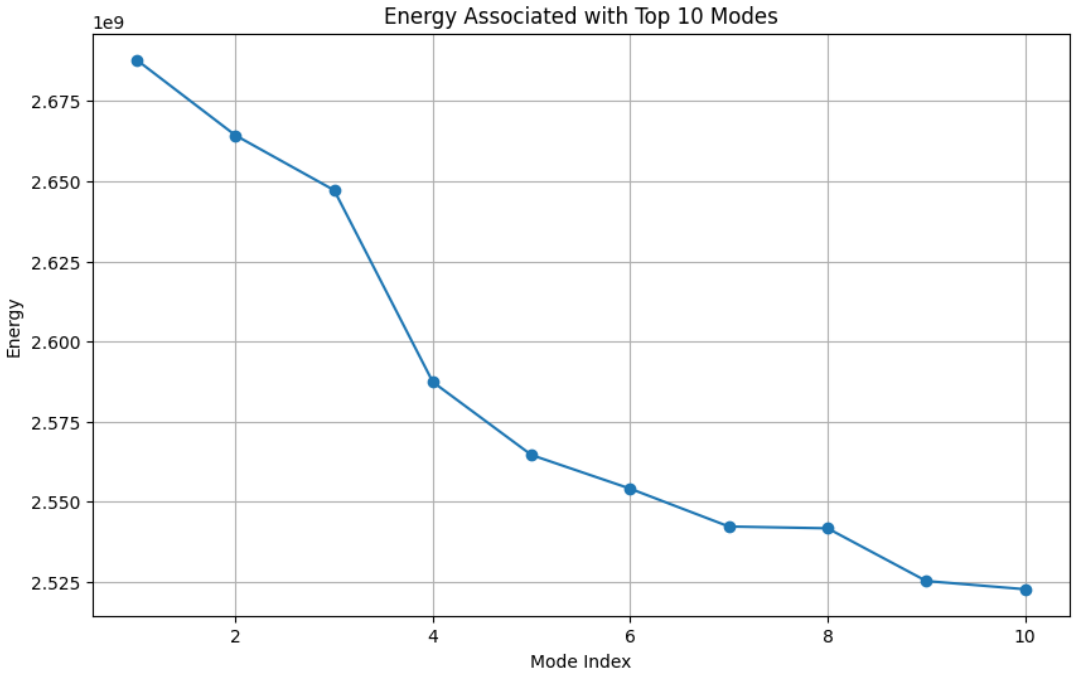


Figure 26: Top 10 energy modes for 60 percent of gaussian noise

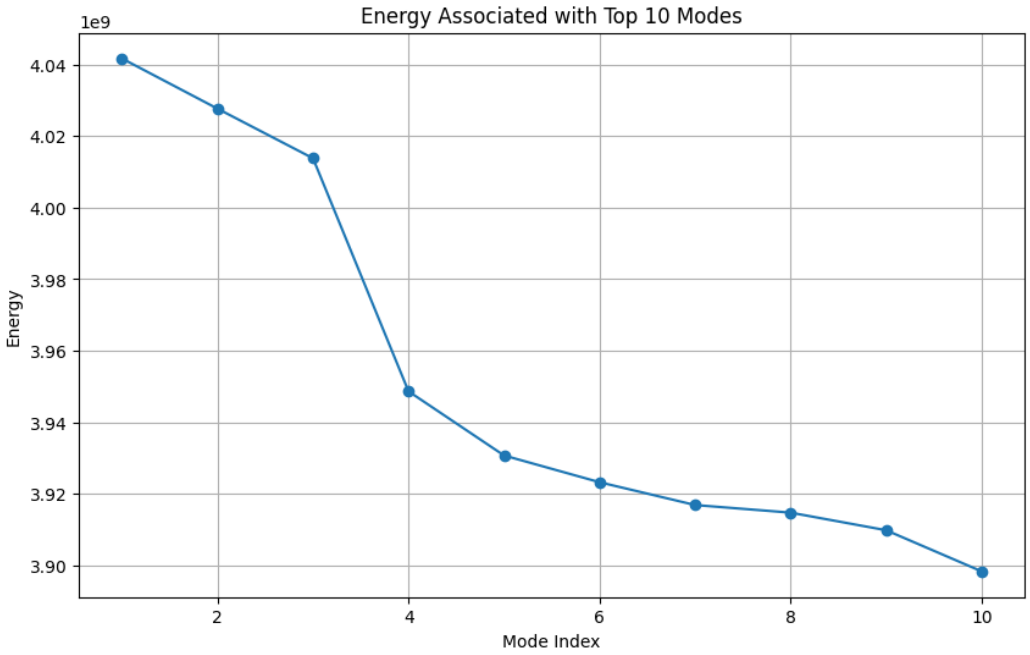


Figure 27: Top 10 energy modes for 80 percent of gaussian noise

As we can observe from the figure 3 the energy of real mode was 352374824.5365
On comparing the energies that we get from super resolving are 304000770.3912 ,302862050.6820 , 300150092.0330 and 294103045.1103 for 20, 40 , 60 and 80 percent noise respectively. Is approximately same real energy i.e. 352374824.5365 . So we can say that our denoising model has given the best accuracy in all cases with little bit variation.

6 References

- **Machine Learning for Fluid Mechanics**
  Steven L. Brunton, Bernd R. Noack and Petros Koumoutsakos4
  https://arxiv.org/pdf/1905.11075.pdf

- **Enhancing Computational Fluid Dynamics with Machine Learning**
  Ricardo Vinuesa and Steven L. Brunton
  https://arxiv.org/pdf/2110.02085v2.pdf