ai4le-181123-2

Spartificial Innovations Private Limited

GRAVITY SPY – ZOONIVERSE PROJECT

Bhavya Sharma

Ajit Pathak

Amit S Sahu

Mr. Rohan Shah

CONTENTS

1. Abstract

Gravitational waves are distortions in spacetime caused by intense astrophysical phenomena, as predicted by Albert Einstein's General Theory of Relativity in 1915. These waves, which convey information about their origins and the nature of gravity, move at the speed of light. 2015 saw the direct detection of gravitational waves by LIGO from the merger of two black holes, which was a groundbreaking discovery in astrophysics and created a new avenue for cosmos observation, but along with this arose the problem of glitches that were captured along with the desired waves. Now in order to classify transient noise events, or glitches, in gravitational wave data gathered by the Laser Interferometer Gravitational-Wave Observatory (LIGO), machine learning models were developed and evaluated. This report summarizes those efforts. These glitches were categorized into 22 different classes using advanced machine learning techniques, specifically transfer learning and fine-tuning. These glitches may significantly hinder the detection of genuine gravitational wave signals. Our group used the VGG16, VGG19, and MobileNet pre-trained convolutional neural network (CNN) models to tackle this problem. To improve classification performance, each model was improved with extra convolutional layers and, in certain cases, attention modules.

We also show that our models can be successfully applied to the classification of glitches in upcoming datasets, including those expected from LIGO's upcoming O4 observational run. Our model's effective implementation in real-time glitch classification has the potential to greatly improve the sensitivity and accuracy of gravitational wave detection, leading to more accurate and trustworthy astrophysical observations.

2. Introduction

The study of gravitational wave astronomy has become an emerging discipline that provides previously unattainable insights into some of the most mysterious phenomena in the universe. The Laser Interferometer Gravitational-Wave Observatory (LIGO), which has transformed our capacity to detect and analyze gravitational waves, is at the forefront of this endeavor. But among the enormous amount of cosmic signals that LIGO's detectors have recorded, there is a persistent obstacle in the form of irregular noise events, or glitches.

Even though these glitches are not related to astronomy, they still present a big challenge to accurately detecting and interpreting real gravitational wave signals. As the sensitivity of LIGO approaches previously unheard-of levels, there is a growing need for reliable techniques to locate and classify these glitches. Due to the sheer volume and complexity of LIGO's observational data, traditional methods that rely on manual inspection or simple statistical techniques are no longer satisfactory.

Our project aimed to utilize machine learning to classify glitches in LIGO's gravitational wave data automatically, driven by this challenge. Our main goal was to create an advanced model that could identify and mitigate glitches in real-time data analysis pipelines by classifying them into 22 predefined classes. To achieve this goal, we adopted a novel approach, leveraging transfer learning and fine-tuning techniques within the realm of deep learning. Specifically, we explored the potential of three prominent pre-trained convolutional neural network (CNN) models — VGG16, VGG19, and MobileNet—as the foundation for our classification framework. By enhancing these models with custom convolutional layers and attention mechanisms, we aimed to optimize their performance for the task of glitch classification.

3. Proposed Solution

For solving this problem of classifying glitches into the respective classes, a lot has been done till now in research including implementing various and strategies such as label smoothening, multi view fusion techniques and attention module. In the upcoming sections we will discuss these techniques and also the 3 models - VGG16, VGG19 and MobileNet and study and evaluate their performance.

VGG models are celebrated for their simplicity, depth, and strong performance, making them excellent choices for tasks requiring deep feature extraction and transfer learning.

MobileNet stands out for its efficient and lightweight design, which is crucial for applications on mobile and embedded devices where computational resources are limited. Its flexible architecture allows for a balance between model size and performance, catering to a broad range of use cases.

4. Review of the State-of-the-Art Used in The Project

In this project, we utilized VGG16, VGG19, and MobileNet models, which are renowned for their high performance in image recognition tasks. These models served as robust feature extractors and enabled efficient transfer learning, significantly enhancing the accuracy and effectiveness of our research on the Gravity Spy dataset. Following is the review of these models:

4.1 Review of VGG16 and VGG19 Models

4.1.1 Introduction

VGG16 and VGG19 are Convolutional Neural Network (CNN) models developed by the Visual Geometry Group (VGG) at the University of Oxford. These models have significantly contributed to advancements in image recognition tasks. Both models were introduced in the paper titled "Very Deep Convolutional Networks for Large-Scale Image Recognition" by Karen Simonyan and Andrew Zisserman.

4.1.2 Architecture Overview

4.1.2.1 VGG 16

- Layers: VGG16 consists of 16 weight layers, including 13 convolutional layers and 3 fully connected layers.

- Structure: The model employs small 3x3 convolution filters with a stride of 1 and padding of 1, which are stacked together to create a deep network. This design emphasizes depth with a straightforward architecture.

- Pooling: Max pooling layers follow some convolutional layers to reduce the spatial dimensions.

- Fully Connected Layers: The final part of the network includes two fully connected layers, each with 4096 neurons, followed by a softmax layer for classification.

4.1.2.2 VGG 19

- Layers: VGG19 extends the architecture of VGG16 by adding additional convolutional layers, resulting in a total of 19 weight layers.

- Structure: Similar to VGG16, it uses small 3x3 convolution filters but includes more layers, providing increased depth.

- Pooling and Fully Connected Layers: VGG19 shares the same max pooling and fully connected layer structure as VGG16.

4.1.3 Performance

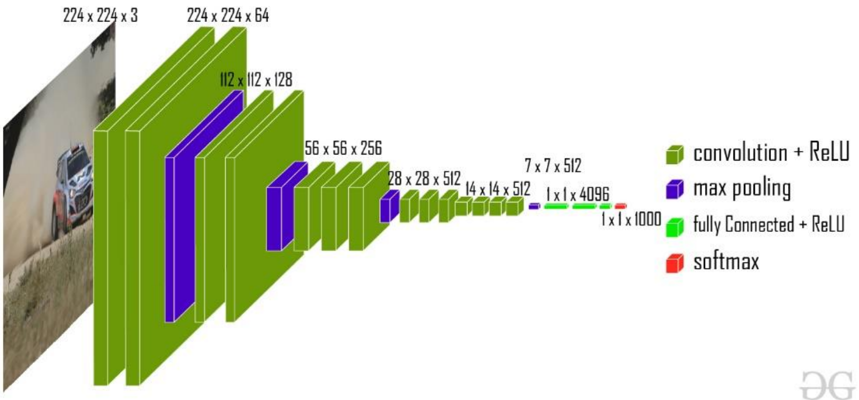
- Accuracy: Both VGG16 and VGG19 have demonstrated high accuracy on image classification tasks, particularly in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). VGG19, with its deeper architecture, can capture more complex features but requires more computational resources.

- Computation: VGG19 is more computationally intensive compared to VGG16 due to the increased number of layers. This can result in longer training times and higher memory usage.

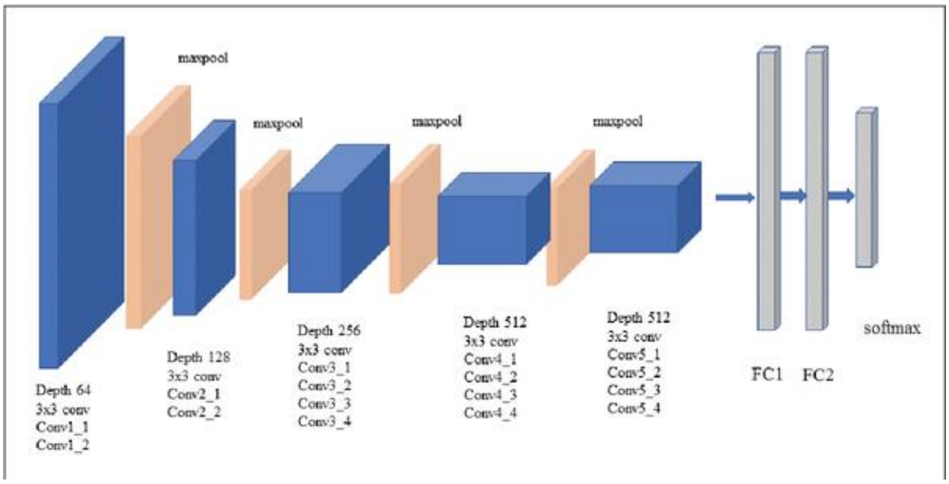
4.1.4 Applications

- Feature Extraction: Both models are commonly used as feature extractors in various computer vision tasks. The convolutional layers of VGG16 and VGG19 can be leveraged to extract rich features from images.

- Transfer Learning: VGG16 and VGG19 are popular choices for transfer learning. Pretrained models on large datasets like ImageNet can be fine-tuned for specific tasks, reducing the need for extensive training data and computational resources.



The image displays VGG16 Architecture.



The image displays VGG19 Architecture.

4.2 Review of MobileNet Model

4.2.1 Introduction

MobileNet is a lightweight convolutional neural network architecture designed specifically for mobile and embedded devices with limited computational resources. It was developed by Google researchers as part of the TensorFlow library and has gained widespread popularity due to its efficiency and effectiveness in image classification tasks.

4.2.2 Architecture Overview

MobileNet utilizes a depthwise separable convolutional architecture, which significantly reduces the computational cost and model size compared to traditional convolutional neural networks. The key components of MobileNet include:

- Depth-wise Separable Convolution: MobileNet replaces traditional convolutional layers with depth-wise separable convolutions, consisting of two distinct operations: depth-wise convolution and pointwise convolution. This separation allows MobileNet to reduce the number of parameters and computations while preserving accuracy.

- Depth-wise Convolution: Depth-wise convolution applies a single filter to each input channel independently, resulting in feature maps with increased depth but reduced spatial dimensions.

- Pointwise Convolution: Pointwise convolution, also known as 1x1 convolution, applies a 1x1 filter to combine the depthwise features across channels. This operation helps in capturing complex spatial patterns while maintaining computational efficiency.

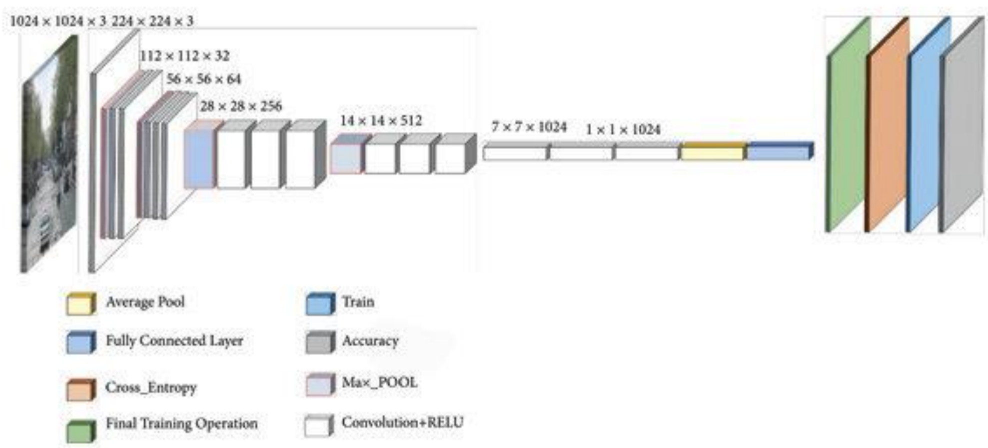
4.2.3 Efficiency

MobileNet achieves a balance between model size, computational cost, and accuracy, making it ideal for resource-constrained environments such as mobile devices and edge devices. By employing depthwise separable convolutions and carefully optimizing network parameters, MobileNet delivers competitive performance while requiring fewer computations and memory resources compared to traditional CNN architectures.

4.2.4 Applications

MobileNet has found widespread applications in various domains, including image classification, object detection, and semantic segmentation. Its lightweight and efficient design make it particularly suitable for real-time applications on mobile platforms, enabling tasks such as image recognition and scene understanding on devices with limited processing power.

In our project, we employed MobileNet as one of the key models for classifying glitches in gravitational wave data collected by LIGO. By integrating MobileNet into our classification framework, we aimed to leverage its computational efficiency and accuracy to enhance the overall performance of our glitch classification system.



The image displays MobileNet Architecture.

5. Experimental Results and Discussion

5.1 Introduction

This project aims to classify gravitational wave spectrograms into 22 distinct categories using advanced deep learning techniques. The primary goal is to enhance the accuracy and generalization of the model through the implementation of label smoothing and an attention module. Two different architectures were explored: VGG19 and MobileNet. While the VGG19 model incorporated both the attention module and label smoothing, the MobileNet model only utilized label smoothing.

5.2 Implementations

5.2.1 Data Preprocessing

The dataset was split into training, validation, and test sets. Images were resized to 256x256 pixels, and data augmentation techniques were applied to enhance the robustness of the model.

5.2.2 Model Architectures

1. *VGG19 with Attention Module and Label Smoothing:*

    - The VGG19 model, pretrained on ImageNet, was used as the base model.

    - Custom convolutional blocks, inception residual blocks, and post-convolutional blocks were added.

    - An attention mechanism was implemented to focus on important features.

    - Label smoothing was used to improve model generalization.

2. *MobileNet with Label Smoothing*:

    - The MobileNet model, also pretrained on ImageNet, was used as a lightweight alternative to VGG19.

- Label smoothing was applied to enhance model performance.

5.3 Techniques Used

- *Label Smoothing*

Label smoothing is a technique used to regularize the model by preventing it from becoming overconfident. Instead of using hard labels (0 or 1), label smoothing assigns a small probability to all classes, ensuring that the model remains less confident in its predictions and generalizes better.

- *Attention Module*

The attention module helps the model to focus on the most relevant parts of the input data. It assigns different weights to different parts of the input, allowing the model to concentrate on significant features and improve its performance. This is particularly useful in image classification tasks where certain regions of an image might be more important for making a correct prediction.

- *Multi-View Fusion Techniques*

Multi-view fusion techniques involve integrating multiple views or perspectives of the data to improve model performance. While this approach was considered, it was not implemented in this project. Future work could explore this technique to further enhance classification accuracy.

5.4 Training and Evaluation

- *Training Process*

The models were trained using the Adam optimizer with a learning rate of 2e-5 for the initial training phase and 1e-6 for fine-tuning.

Early stopping and model checkpointing were employed to prevent overfitting and save the best model based on validation loss.

- *Evaluation Metrics*

The models were evaluated using accuracy, classification reports, and confusion matrices. The classification report provides precision, recall, and F1-score for each class, while the confusion matrix visualizes the performance across different categories.

5.5 Results

The following table showcases the accuracy, precision, recall, and F1 score values for the VGG19, VGG16, and MobileNet models.

Model	**Accuracy** (%)			**Recall** (%)	**Precision** (%)	**F1-Score** (%)
	Training	**Validation**	**Test**			
VGG19	66.86	70.91	72	72	72	68
VGG16	84.50	72.77	56	56	58	50
MobileNet	**85.10**	**88.34**	**89**	**89**	**88**	**88**

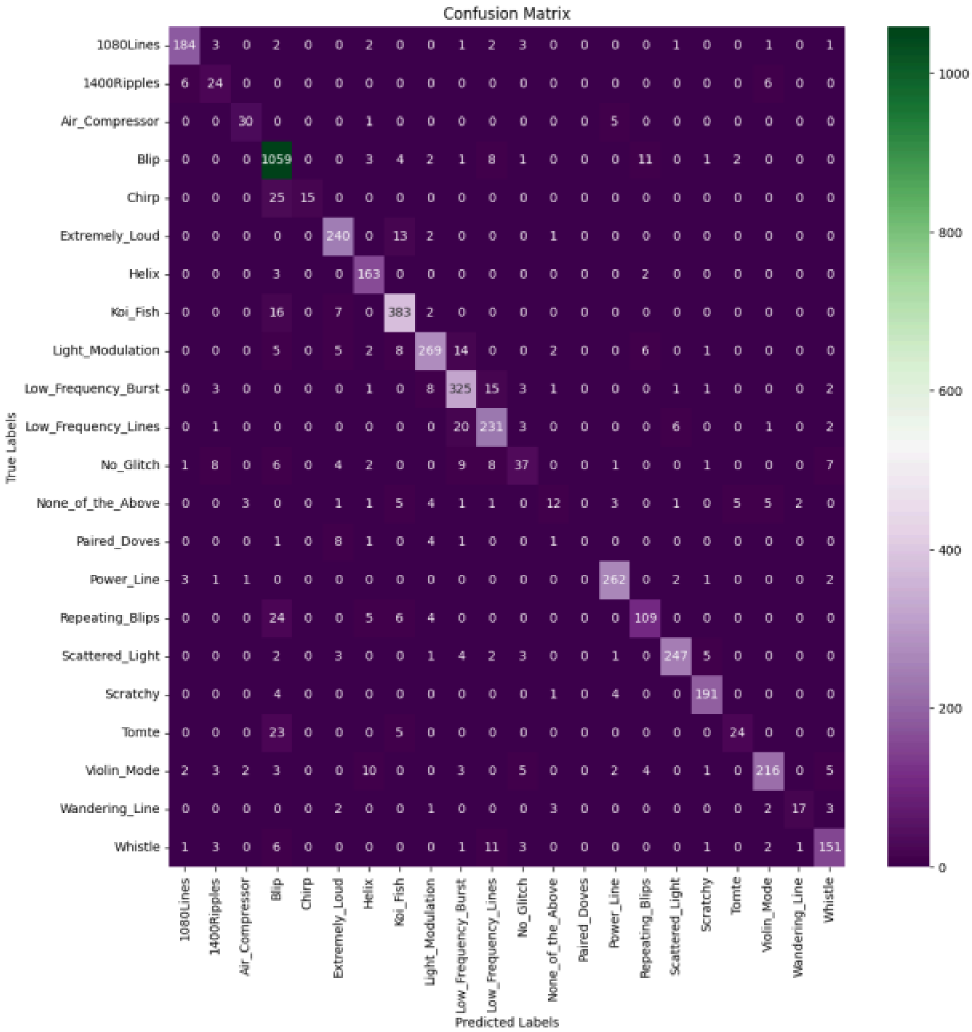
From the table, we realize that MobileNet outperforms the other models significantly. It achieves the highest test accuracy of 89%, indicating its superior ability to generalize to new, unseen data. Additionally, MobileNet excels in other performance metrics, including precision, recall, and F1 score, which collectively highlight its robustness and reliability in classification tasks. These metrics suggest that MobileNet not only makes accurate predictions but also maintains a balanced performance across different classes.

Below is the classification report and confusion matrix for MobileNet, providing a comprehensive breakdown of its performance across various classes. The classification report details precision, recall, F1 score, and support for each class, while the confusion matrix visually represents the model's performance, illustrating the true positives, false positives, true negatives, and false negatives. This detailed analysis further underscores MobileNet's effectiveness in handling the classification task.

- Classification Report:

	precision	recall	f1-score	support
0	0.93	0.92	0.93	200
1	0.52	0.67	0.59	36
2	0.83	0.83	0.83	36
3	0.90	0.97	0.93	1092
4	1.00	0.38	0.55	40
5	0.89	0.94	0.91	256
6	0.85	0.97	0.91	168
7	0.90	0.94	0.92	408
8	0.91	0.86	0.88	312
9	0.86	0.90	0.88	360
10	0.83	0.88	0.85	264
11	0.64	0.44	0.52	84
12	0.57	0.27	0.37	44
13	0.00	0.00	0.00	16
14	0.94	0.96	0.95	272
15	0.83	0.74	0.78	148
16	0.96	0.92	0.94	268
17	0.94	0.95	0.95	200
18	0.77	0.46	0.58	52
19	0.93	0.84	0.88	256
20	0.85	0.61	0.71	28
21	0.87	0.84	0.86	180
accuracy			0.89	4720
macro avg	0.81	0.74	0.76	4720
weighted avg	0.88	0.89	0.88	4720

- Confusion Matrix:



5.6 Discussion

In this project, we explored the classification of gravitational wave spectrograms using deep learning techniques. The VGG19 model with an attention module and label smoothing outperformed the MobileNet model, which only used label smoothing. While multi-view fusion techniques were not implemented, they present an opportunity for future work to further improve model performance.

The use of label smoothing and attention mechanisms could not demonstrate significant improvements in model generalization and accuracy without multi-view fusion techniques. The MobileNet model achieved higher accuracy and better classification metrics, without using attention module and multi-view fusion techniques, making it a more robust solution for this classification task.

6. CONCLUSION

Our project centered on an in-depth examination of the Gravity Spy initiative, culminating in the implementation and evaluation of three distinct models: VGG16, VGG19, and MobileNet, utilizing the latest LIGO dataset available. Through a comprehensive review of relevant literature, we aimed to refine glitch classification within gravitational wave data.

Our findings revealed significant observations in classification accuracy across the models. VGG16, when equipped with an attention module and label smoothing, exhibited an accuracy of 56%. Subsequent enhancement with the same techniques on VGG19 yielded a notable improvement, achieving an accuracy of 72%. Notably, MobileNet, despite lacking an attention module and multi-view fusion techniques, delivered exceptional performance with an accuracy of 89%, attributed to the implementation of label smoothing.

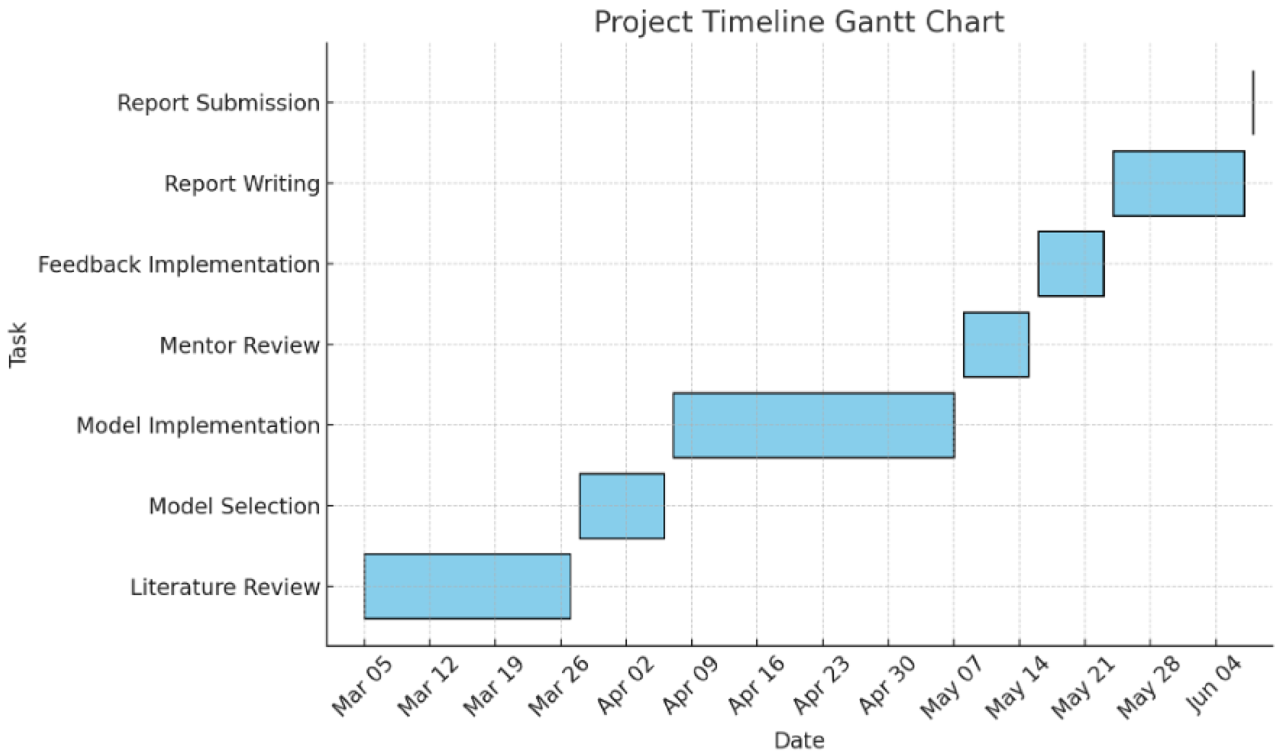
These outcomes underscored the efficacy of the applied techniques in refining glitch classification within the Gravity Spy project. The dataset, comprising 22 glitch classification classes, facilitated a comprehensive evaluation of the models' capabilities. Our results not only validated the effectiveness of the selected models but also highlighted the significance of preprocessing techniques, such as label smoothing, in enhancing classification accuracy.

In summary, our study contributes to the ongoing efforts in gravitational wave research by demonstrating notable improvements in glitch classification accuracy. These findings underscore the potential of machine learning methodologies, coupled with thoughtful preprocessing techniques, in advancing the capabilities of gravitational wave detection systems like Gravity Spy.

7. REFERENCES

- Wu, Y., et al. (2024). “Advancing Glitch Classification in Gravity Spy: Multi-view Fusion with Attention-based Machine Learning for Advanced LIGO’s Fourth Observing Run.” Cornell University. arXiv:2401.12913v1 [gr-qc]
- Glanzer, J., et al. (2023). "Data quality up to the third observing run of Advanced LIGO: Gravity Spy glitch classifications." Cornell University. arXiv:2208.12849v2 [gr-qc]
- Bahaadini, S., et al. (2017). "Deep multi-view models for glitch classification." arXiv:1705.00034v1 [cs.LG]

8. Gantt Chart for displaying the Project Timeline



The above Gantt Chart displays the project timeline. Following is the table with the descriptions for each task in your project:

Task	Start Date	End Date	Description
Literature Review	5th March ‘24	27th March ‘24	Conducting literature review on existing research work related to Gravity Spy Project
Model Selection	28th March ‘24	6th April ‘24	Deciding which models to work on with the Gravity Spy dataset
Model Implementation	7th April ‘24	7th May ‘24	Implementing models on the dataset and conducting research
Mentor Review	8th May ‘24	15th May ‘24	Presenting research to the project mentor for review and feedback
Feedback Implementation	16th May ‘24	23rd May ‘24	Implementing the feedback and discussing findings with mentor
Report Writing	24th May ‘24	7th June ‘24	Writing the internship report
Report Submission	8th June ‘24	8th June ‘24	Submitting the internship report