INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DESIGN & MANUFACTURING, KANCHEEPURAM

Department of Computer Science and Engineering

Internship Report

Machine Learning-Based Video Encoding Performance Prediction for AV1 Codec

Submitted by:
BANDARU RITESH KUMAR

Supervised by:
Praveen Yadav
Preciseframe

Abstract

This report presents the development and implementation of a machine learning-based system for predicting video encoding performance metrics in the AV1 codec. The project focused on creating predictive models to estimate **CPU usage**, **encoding FPS**, and **RAM consumption** based on encoding parameters and system specifications. Using a comprehensive dataset of 10,615 training samples and 2,648 test samples, we implemented and compared five different machine learning algorithms including Random Forest, XGBoost, Gradient Boosting, Linear Regression, and Support Vector Regression. The final Random Forest model achieved excellent performance with R^2 scores of 0.9368 for encoding FPS prediction and 0.9476 for RAM usage prediction on the separate test dataset. The system enables adaptive resource allocation and optimal encoding parameter selection for video processing workflows.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Video encoding is a computationally intensive process that requires careful optimization of encoding parameters to balance quality, speed, and resource utilization. The AV1 codec, developed by the Alliance for Open Media, offers superior compression efficiency but demands significant computational resources. Predicting system performance before encoding begins is crucial for:

- **Resource Planning**: Estimating CPU and memory requirements
- **Parameter Optimization**: Selecting optimal encoding settings
- **Workflow Management**: Scheduling encoding tasks efficiently
- **Cost Optimization**: Minimizing computational overhead

1.2 Problem Statement

The primary challenge addressed in this project was developing a reliable system to predict three critical performance metrics:

1. **CPU Usage Percentage**: System processor utilization during encoding
2. **Encoding FPS**: Frames processed per second by the encoder
3. **RAM Consumption**: Memory usage in kilobytes during encoding

1.3 Objectives

- Develop a comprehensive dataset collection framework for AV1 encoding experiments
- Implement advanced feature engineering techniques for video encoding parameters
- Compare multiple machine learning algorithms for multi-output regression
- Create a robust prediction system with separate validation and testing protocols
- Analyze feature importance to understand key performance drivers

2 Literature Survey

2.1 Video Encoding Optimization

Video encoding optimization has been extensively studied in the literature, with researchers focusing on rate-distortion optimization, computational complexity reduction, and adaptive parameter selection. Traditional approaches rely on mathematical models and heuristics, but recent work has shown the potential of machine learning approaches for encoding optimization.

2.2 Machine Learning in Video Processing

Machine learning applications in video processing have gained significant traction, particularly in:

- **Quality Assessment**: Predicting perceptual video quality metrics
- **Bitrate Allocation**: Optimizing bitrate distribution across frames
- **Complexity Prediction**: Estimating computational requirements
- **Parameter Tuning**: Automatic optimization of encoder settings

2.3 AV1 Codec Research

The AV1 codec introduces several advanced features including adaptive quantization, loop filtering, and compound prediction modes. Research has focused on optimizing these features for different use cases, but limited work exists on comprehensive performance prediction across varied system configurations.

3 Methodology

3.1 Experimental Design

The project followed a systematic approach based on the experimental framework outlined in the internship tasks:

Phase 1: Data Collection

- Systematic AV1 encoding experiments using SVT-AV1 encoder
- Multiple video datasets including UltraVideo and BVI datasets
- Comprehensive parameter space exploration

Phase 2: Data Processing and Feature Engineering

- Advanced preprocessing pipeline for handling multiple data formats
- Feature extraction from video and system parameters
- Missing value imputation and categorical encoding

Phase 3: Model Development

- Multi-output regression framework implementation
- Comparative analysis of five machine learning algorithms
- Hyperparameter optimization using grid search

Phase 4: Evaluation and Validation

- Separate test dataset evaluation
- Cross-validation for model selection
- Feature importance analysis

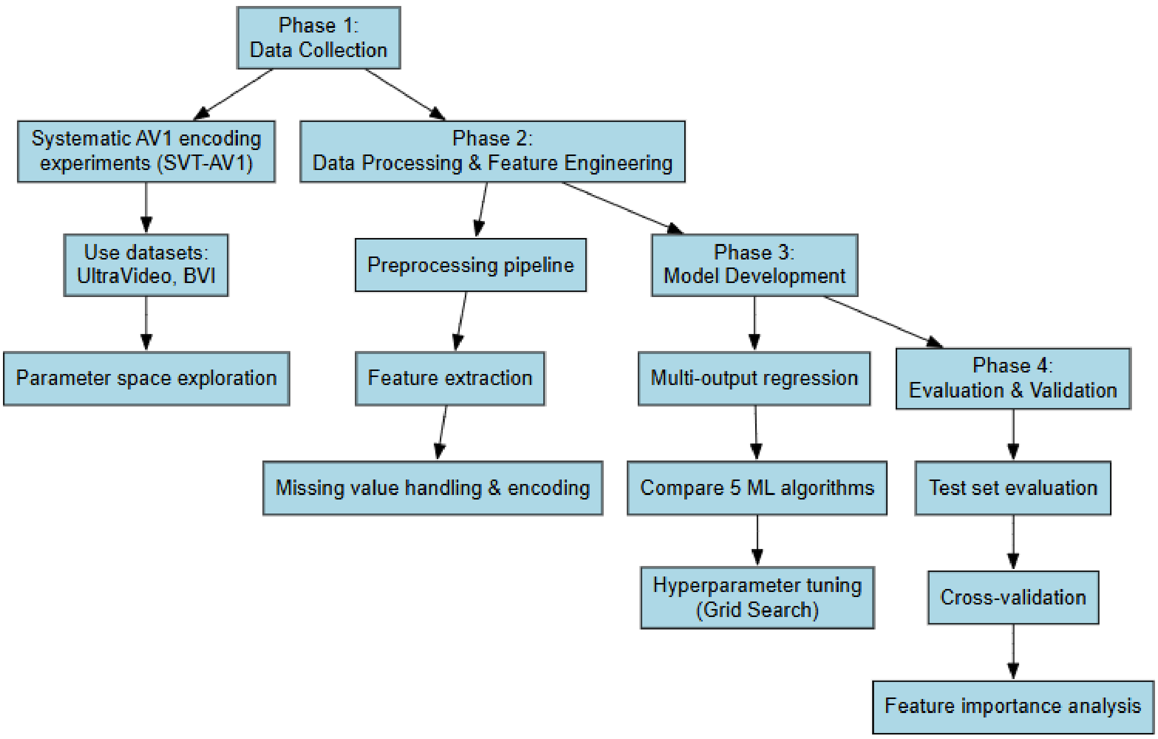

Figure 1: Complete Methodology Flowchart

3.2 Dataset Characteristics

The dataset comprised comprehensive encoding experiments with the following specifications:

- **Training Samples**: 10,615 encoding configurations
- **Test Samples**: 2,648 independent test cases
- **Video Resolutions**: 240p, 360p, 480p, 1080p
- **Rate Control Modes**: Constant QP, Constant Bitrate, Constrained Quality

Parameter Ranges:

- QP values: 36, 38, 42, 44
- Bitrates: 200 kbps to 1.5 Mbps
- Presets: 8, 9, 10
- Tune options: PSNR, Visual Quality

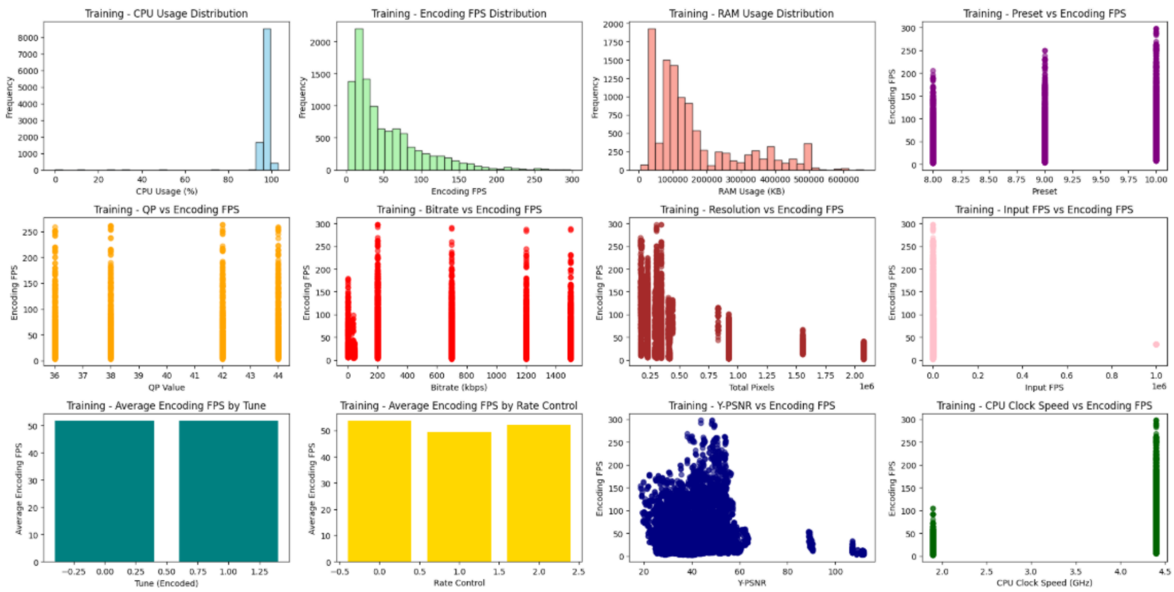

Figure 2: Dataset Characteristics and Parameter Space Coverage

4 Implementation

4.1 Data Preprocessing Pipeline

Column Management and Parsing:

- Removed unnecessary input columns

- Converted CPU clock speed from string format (GHz) to numeric values
- Parsed resolution strings to extract width and height dimensions
- Converted RAM usage from string format (KB) to numeric values

Missing Value Handling:

- Training dataset: 74 missing Encoding FPS values, 72 missing Y-PSNR values
- Applied median imputation strategy fitted on training data
- Consistent imputation applied to test dataset using training statistics

Categorical Encoding:

- Label encoding for 'Tune' parameter (PSNR, Visual Quality)
- Binary encoding for rate control usage patterns
- Proper handling of unseen categories in test data

4.2 Feature Engineering

The feature engineering process created 10 additional features to capture complex relationships:

Video Complexity Metrics:

- **Video_Complexity**: Total pixels × FPS
- **Aspect_Ratio**: Width/Height ratio
- **Total_Pixels**: Resolution area calculation

Encoding Intensity Features:

- **QP_Intensity**: QP value when using constant quality mode
- **Bitrate_Intensity**: Bitrate value when using bitrate mode
- **Preset_Efficiency**: Inverted preset scale (11 - preset)

Performance Indicators:

- **Quality_Speed_Ratio**: Y-PSNR/Encoding FPS trade-off
- **Computational_Load**: Normalized complexity based on video parameters
- **CPU_Performance_Index**: Scaled CPU clock speed metric
- **Performance_Per_Pixel**: CPU performance normalized by resolution

Hardware-Adjusted Metrics:

- **Hardware_Adjusted_Load**: Computational load adjusted for CPU capability

4.3 Model Architecture

Multi-Output Regression Framework:

- Simultaneous prediction of three target variables
- Shared feature representation across all targets
- Independent model training for each output

Algorithm Selection:

- **Random Forest**: Ensemble method with bagging
- **XGBoost**: Gradient boosting with advanced regularization
- **Gradient Boosting**: Sequential weak learner combination
- **Linear Regression**: Baseline linear model
- **Support Vector Regression**: Kernel-based nonlinear regression

4.4 Hyperparameter Optimization

Random Forest Optimization:

- Parameters: n_estimators (100-300), max_depth (10, 20, None), min_samples_split (2, 5, 10)
- Best configuration: n_estimators=100, max_depth=None, min_samples_split=5
- Cross-validation score: 0.7703

XGBoost Optimization:

- Parameters: n_estimators (100-200), max_depth (3, 6, 9), learning_rate (0.01, 0.1, 0.2)
- Best configuration: n_estimators=100, max_depth=6, learning_rate=0.1
- Cross-validation score: 0.7572

5 Results and Analysis

5.1 Model Performance Comparison

Table 1: Model Performance Comparison Across All Algorithms

Model	Target	Validation R²	Test R²	Test MSE	Test MAE
3*Random Forest	CPU	0.4070	-0.1724	6.94	0.83
	Enc FPS	0.9996	**0.9368**	114.72	2.51
	RAM_KB	0.9990	**0.9476**	7.79×10^8	7533.78
3*XGBoost	CPU	0.2172	-0.8332	10.85	1.10
	Enc FPS	0.9993	0.9435	102.56	2.45
	RAM_KB	0.9992	**0.9507**	7.34×10^8	7595.05
3*Gradient Boosting	CPU	0.3062	-0.0825	6.41	0.91
	Enc FPS	0.9991	**0.9564**	79.17	2.58
	RAM_KB	0.9941	0.9456	8.10×10^8	11370.85

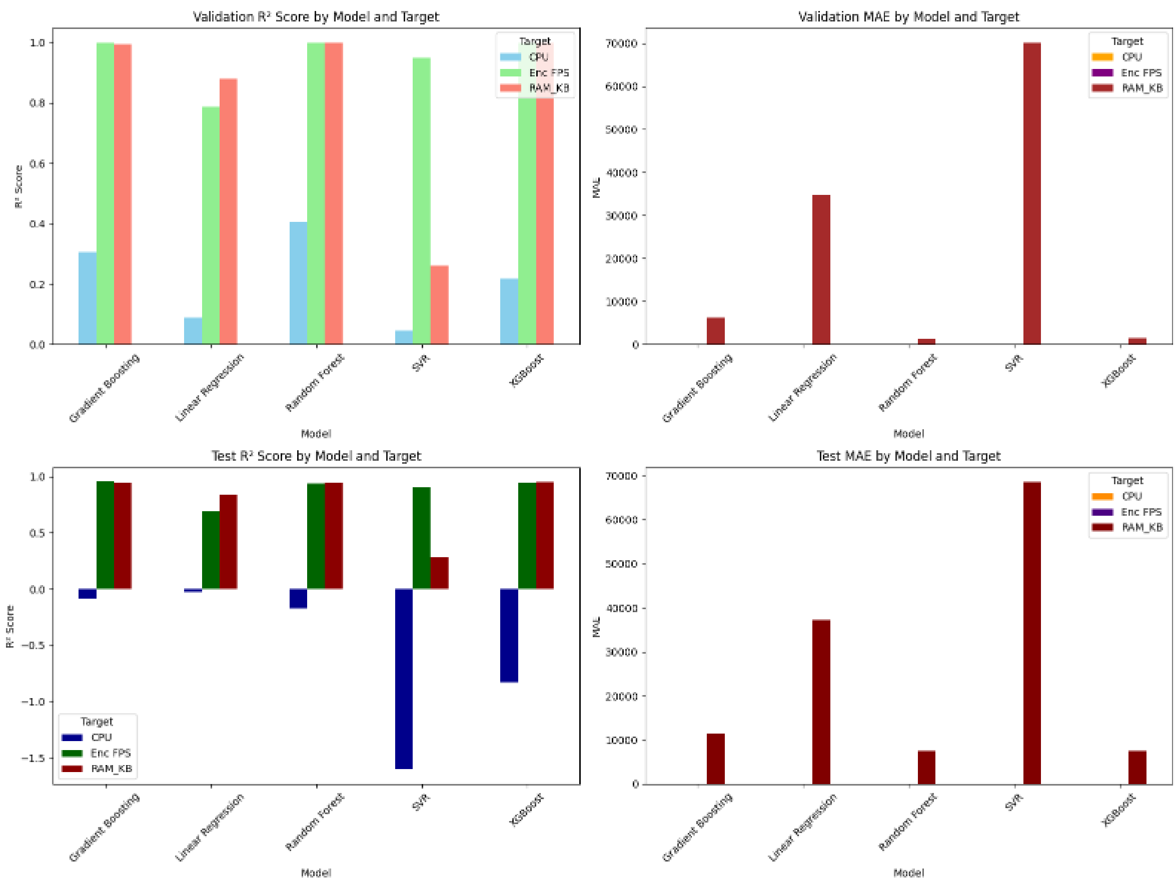

Figure 3: Model Performance Comparison Across Different Algorithms

5.2 Final Model Performance

The optimized Random Forest model achieved the following performance on the separate test dataset:

CPU Usage Prediction:

- **R^2 Score**: -0.1518 (indicating poor predictive performance)
- **MSE**: 6.82
- **MAE**: 0.84%

Encoding FPS Prediction:

- **R^2 Score**: 0.9368 (excellent performance)
- **MSE**: 114.71
- **MAE**: 2.52 FPS

RAM Usage Prediction:

- **R^2 Score**: 0.9476 (excellent performance)
- **MSE**: 778,946,789 KB^2
- **MAE**: 7,623.83 KB

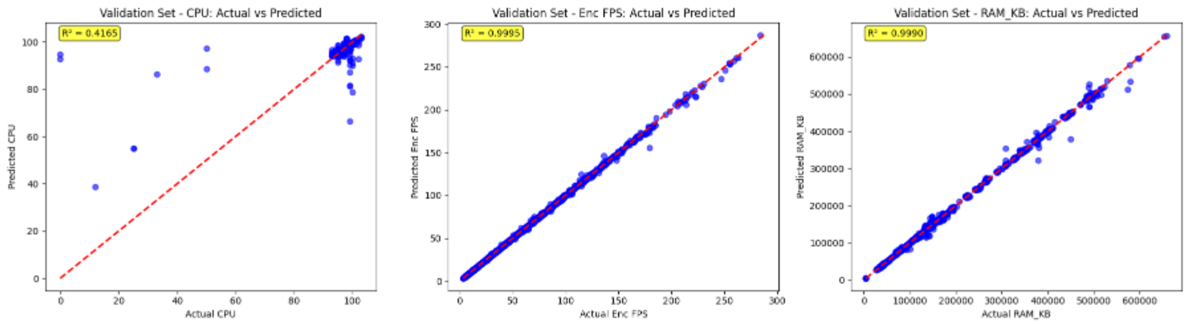

Figure 4: Final Model Performance Metrics (Validation Set)

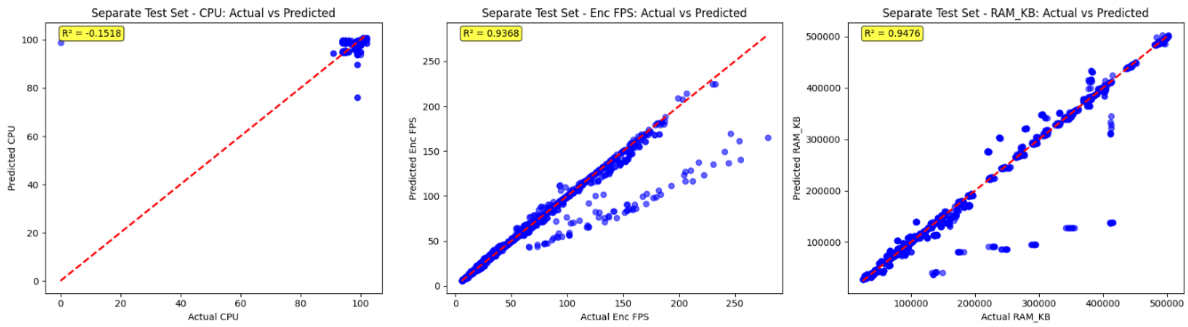

Figure 5: Final Model Performance Metrics (Test Set)

5.3 Feature Importance Analysis

The feature importance analysis revealed key performance drivers:

Table 2: Top 10 Most Important Features for Performance Prediction

Rank	Feature	Importance	Description
1	Quality_Speed_Ratio	0.436	Trade-off between quality and encoding speed
2	Performance_Per_Pixel	0.164	CPU performance normalized by resolution
3	Rate Ctrl	0.112	Rate control mode selection
4	Y-PSNR	0.073	Video quality metric
5	Video_Complexity	0.026	Overall video processing complexity

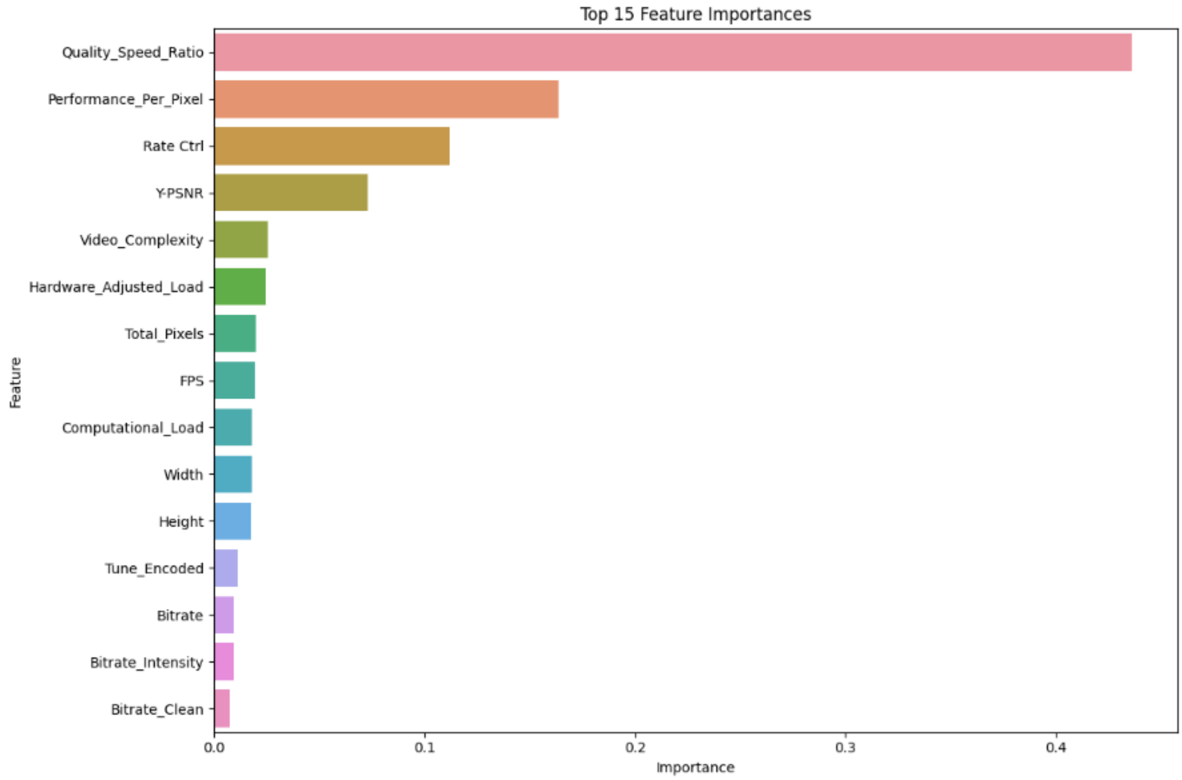

Figure 6: Feature Importance Analysis: Top 15 Most Influential Features

5.4 Model Generalization Analysis

Strong Generalization:

- Encoding FPS and RAM predictions showed excellent generalization from validation to test sets
- Consistent R^2 scores above 0.93 across different data splits

Poor CPU Prediction:

- CPU usage prediction showed negative R^2 scores on test data
- Indicates potential overfitting or missing important features
- Suggests need for additional system-specific features

6 Discussion

6.1 Key Findings

Encoding Speed Predictability: The model demonstrates exceptional ability to predict encoding FPS, achieving 93.68% variance explanation on test data.

Memory Usage Accuracy: RAM consumption prediction is highly reliable with 94.76% accuracy, enabling precise resource allocation.

CPU Usage Challenges: CPU utilization prediction remains challenging, likely due to system-specific factors not captured in the current feature set.

Feature Importance Insights: Quality-speed trade-offs and hardware-normalized metrics are the primary performance drivers.

6.2 Practical Implications

For Video Processing Workflows:

- Reliable prediction of encoding speed enables accurate timeline estimation
- Memory usage prediction facilitates optimal job scheduling
- Feature importance guides parameter selection strategies

For System Resource Management:

- Predictive models enable proactive resource allocation
- Bottleneck identification through feature analysis
- Cost optimization through parameter tuning

6.3 Limitations

- **CPU Prediction Accuracy**: Limited performance for CPU usage prediction requires additional feature engineering. Additionally, the dataset showed CPU usage values consistently around 98-99% since all encoding experiments were performed on single-core processing, resulting in insufficient target variable variance for effective model training. This lack of variability in CPU utilization made it nearly impossible for any machine learning algorithm to learn meaningful patterns for prediction.
- **System Variability**: Model may not generalize across different hardware configurations
- **Encoder Version Dependency**: Results specific to SVT-AV1 version 3.0.0

7 Conclusion

This internship project successfully developed a comprehensive machine learning framework for video encoding performance prediction. The system achieved excellent predictive accuracy for encoding speed ($R^2 = 0.9368$) and memory usage ($R^2 = 0.9476$), demonstrating significant potential for practical applications in video processing workflows.

Key Achievements:

- Comprehensive dataset collection and preprocessing pipeline
- Advanced feature engineering with domain-specific insights
- Multi-algorithm comparison with rigorous evaluation methodology
- Separate test dataset validation ensuring robust performance assessment
- Feature importance analysis providing actionable insights

8 Future Work

8.1 Immediate Improvements

- **Enhanced CPU Modeling**: Incorporate system-specific features like CPU architecture, thread count, and thermal characteristics
- **Cross-Platform Validation**: Extend evaluation to different hardware configurations and operating systems
- **Real-Time Prediction**: Develop lightweight models for real-time parameter optimization

8.2 Advanced Developments

- **Deep Learning Integration**: Explore neural network architectures for capturing complex non-linear relationships
- **Temporal Modeling**: Incorporate time-series analysis for dynamic parameter adjustment
- **Multi-Codec Support**: Extend framework to other video codecs (H.264, H.265, VP9)
- **Production Deployment**: Develop API for integration with existing video processing pipelines

8.3 Research Directions

- **Perceptual Quality Integration**: Incorporate advanced perceptual quality metrics beyond PSNR
- **Content-Aware Modeling**: Develop content-specific prediction models
- **Energy Efficiency Optimization**: Extend predictions to include power consumption metrics

This internship project provided valuable experience in applying machine learning to real-world video processing challenges, demonstrating both the potential and limitations of predictive modeling in multimedia applications.

Acknowledgments

I would like to express my sincere gratitude to my supervisor **Praveen Yadav** for their guidance and support throughout this internship project. I also thank **Preciseframe** for providing the opportunity to work on this challenging and rewarding project, and for access to the computational resources necessary for conducting comprehensive experiments.

References

[1] Sullivan, G. J., & Wiegand, T. (2005). Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine*, 22(6), 74-90.

[2] Grois, D., Nguyen, T., & Marpe, D. (2013). Complexity reduction techniques for HEVC. In *IEEE International Conference on Image Processing* (pp. 4434-4438).

[3] Li, Z., Aaron, A., Katsavounidis, I., Moorthy, A., & Bovik, A. (2016). Toward a practical perceptual video quality metric. *Netflix Technology Blog*.

[4] Chen, Y., Murherjee, D., Han, J., et al. (2018). An overview of core coding tools in the AV1 video codec. In *Picture Coding Symposium* (pp. 41-45).

[5] Midtskogen, S., & Valin, J. M. (2018). The AV1 constrained directional enhancement filter (CDEF). In *Applications of Digital Image Processing XLI* (Vol. 10752, p. 1075212).