Vision-Based Vibration Analysis for Machine Defect Detection using Eulerian Video Magnification and Deep Learning

Mohamed Ahmed (8317), Nour Hassan Ibrahim (8357), Mohamed Emad (7920), Hoor Arafa (7881)
Department of Computer and Communication Engineering
Alexandria University, Egypt

***Abstract*—This paper presents a novel vision-based approach to the detection and classification of mechanical defects using vibration analysis derived from video recordings. Instead of relying on physical sensors, we analyze subtle machine vibrations through video processing and Eulerian Video Magnification (EVM), followed by segmentation and deep feature extraction. The proposed system identifies three machine conditions—Normal State, Bearing Fault, and Unbalanced Weight—using synchronized video streams, temporal processing, and a hybrid of statistical and learning-based classification techniques. Our method achieves up to 99% classification accuracy, demonstrating strong generalization on real-world vibration patterns.**

***Index Terms*—Vibration Analysis, Fault Detection, Machine Vision, Eulerian Video Magnification, Optical Flow, Feature Extraction, Deep Learning.**

I. Introduction

Rotating machinery such as motors and turbines are critical components in industrial systems. Over time, these machines may develop faults due to wear and tear, leading to catastrophic failures if not detected early. Traditional condition monitoring systems rely heavily on accelerometers and specialized sensors, which may be costly, require installation downtime, and are limited in spatial coverage.

In contrast, video-based vibration analysis offers a non-invasive and cost-effective alternative. High-frame-rate cameras can capture machine behavior from a distance, allowing for continuous monitoring and defect detection without disrupting operations. In this paper, we propose a complete pipeline that processes raw video data, amplifies subtle motion using Eulerian Video Magnification (EVM), extracts spatio-temporal features, and classifies machine health using both tabular machine learning models and Convolutional Neural Networks (CNNs).

II. Related Work

Vibration analysis has traditionally relied on time-domain, frequency-domain, and time-frequency methods using data from accelerometers or laser vibrometers. Recent studies have started exploring visual vibration analysis using techniques like Phase-Based Motion Magnification (PBMM) and EVM, pioneered by Wu et al. [1].

Applications of video-based analysis include medical monitoring, structural vibration assessment, and human motion magnification. However, its application in machine fault detection remains underexplored. Our work bridges this gap by integrating video magnification with automated machine learning for real-time diagnostic purposes.

III. System Overview

The pipeline consists of the following major components:

1) Video Collection and Preprocessing
2) Motion Amplification using EVM
3) Temporal Segmentation
4) Frame-wise Feature Extraction
5) Aggregation and Group-wise CSV Generation
6) Machine Learning Classification
7) CNN Frame-Level Classification

Each step is implemented using OpenCV, NumPy, SciPy, and TensorFlow/Keras.

IV. Data Collection and Preprocessing

We recorded six videos representing three machine states:

- **Normal State**
- **Bearing Fault**
- **Unbalanced Weight**

Each condition was recorded from:

- A front view (40 cm distance)
- An angled side view

All videos were captured at 250 RPM and are approximately one minute in length.

A. Video Stabilization

We apply the Farneback dense optical flow algorithm to compute frame-to-frame displacement. Accumulated transformations stabilize the frame sequence, ensuring motion analysis focuses solely on machine behavior and not camera movement.

B. Motion-based ROI Detection

Grayscale frame differences are computed to generate a motion heatmap. We extract contours and calculate a bounding box over regions exhibiting significant motion. This ROI is used for all downstream processing.

C. Equalization and Cropping

Histogram equalization enhances contrast and detail. Each frame is then cropped to the detected ROI and converted to grayscale, ensuring uniformity and relevance.

D. Frame Interpolation and Resizing

To create standardized videos, we interpolate frames to match the maximum frame count across videos. All frames are resized to a common resolution, enabling synchronization during training.

V. Eulerian Video Magnification (EVM)

EVM reveals subtle vibrations by amplifying temporal variations at specific frequency bands. The algorithm involves:

1) Decomposing each frame using a Laplacian pyramid.
2) Applying a temporal bandpass filter (0.1–2.0 Hz).
3) Amplifying and reconstructing magnified frames.

Parameters used:

- Pyramid Levels: 5
- Amplification Factor: 100
- Frequency Band: 0.1–2.0 Hz
- Attenuation: 5

This step transforms imperceptible machine surface vibrations into clearly observable oscillations.

VI. Temporal Segmentation

Each magnified video is segmented into multiple slices:

- 5s, 10s, and 15s duration
- With and without 50% overlap

This produces localized data for time-specific pattern analysis and increases data diversity. Segments are saved in organized directories for systematic labeling.

VII. Frame-Wise Feature Extraction

We extract statistical and signal-based features from each frame:

- **Optical Flow:** Mean and standard deviation of motion vectors.
- **Edge Ratio:** Ratio of edge pixels using Canny detector.
- **Keypoints:** Detected via SIFT.
- **FFT Peak:** Peak frequency magnitude for each frame.

All extracted features are stored in CSV format, annotated by segment ID, view, and machine state.

VIII. Group-wise CSV Aggregation

For each segment configuration (e.g., 10s with overlap), we generate a master CSV:

- Concatenates all frames' features
- Adds metadata such as class, view, duration
- Ready for machine learning pipelines

IX. Model Training and Evaluation

A. Tabular Model Training

We use a fully connected neural network:

- 3 dense layers with ReLU activations
- Dropout and Batch Normalization
- L1/L2 regularization
- Optimizer: Adam with LR scheduling

Training Strategy:

- Stratified 70-15-15 train-val-test split
- Class balancing via weighted loss
- Early stopping and ReduceLROnPlateau

Performance:

- 99% test accuracy with 90% training accuracy
- Minimal overfitting due to strong regularization

B. CNN for Frame Classification

We trained a CNN directly on grayscale magnified frames:

- 3 Conv layers: 8–16–32 filters
- MaxPooling, BatchNorm, Dropout
- Fully connected layer for classification

The CNN achieved 94.37% accuracy on the test set, showing strong discriminative power on visual features.

X. Results and Discussion

Our experiments show that:

- EVM amplifies subtle vibrations that are crucial for visual classification.
- Optical flow and FFT-based features are highly informative.
- Segmentation improves robustness and generalization.
- CNNs perform nearly as well as tabular models without requiring feature engineering.

Both front and angle views contributed to improved accuracy, particularly in ambiguous cases like light bearing faults.

XI. Conclusion

We developed an end-to-end video-based system for machine defect detection without physical sensors. The combination of video magnification, statistical features, and deep learning models achieves high classification accuracy across diverse machine conditions.

Future Work:

- Real-time inference pipeline
- Extended fault classes
- Integration with industrial IoT platforms

Acknowledgments

We would like to thank our professors and mentors at Alexandria University, Department of Computer and Communication Engineering, for their guidance throughout this project.

References

[1] H.-Y. Wu et al., "Eulerian Video Magnification for Revealing Subtle Changes in the World," ACM Trans. Graph., vol. 31, no. 4, 2012.
[2] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.
[3] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," Int. J. of Computer Vision, 2004.
[4] F. Chollet, "Keras: Deep Learning library for Python," GitHub repository, 2015.