Vowel Classification Using KNN and Formant Analysis

1. Introduction

Vowel classification is an essential aspect of speech recognition, leveraging acoustic properties such as formant frequencies and fundamental frequency (F0) for differentiation. This study explores a K-Nearest Neighbors (KNN) based approach to classify five vowel sounds (/a/, /e/, /i/, /o/, /u/) using formant analysis. The objective is to assess the effectiveness of formant-based classification and identify potential improvements for accuracy enhancement.

---

2. Methodology

2.1 Dataset Collection and Storage

The dataset consists of vowel recordings stored in WAV format. The dataset comprises 180 samples, with filenames labeled according to vowel categories (e.g., ‘a_’ for /a/ vowels). The dataset undergoes preprocessing to extract key acoustic features.

2.2 Feature Extraction

The following features were extracted from the audio signals:

- **Formants (F1, F2, F3):** Extracted using Linear Predictive Coding (LPC), providing insights into vowel articulation.
- **Fundamental Frequency (F0):** Estimated using the autocorrelation method, representing the speaker's pitch.

2.3 Data Preprocessing

- Extracted features were normalized using StandardScaler for uniformity.
- The dataset was split into training (80%) and testing (20%) subsets using stratified sampling.

2.4 Classification Using KNN

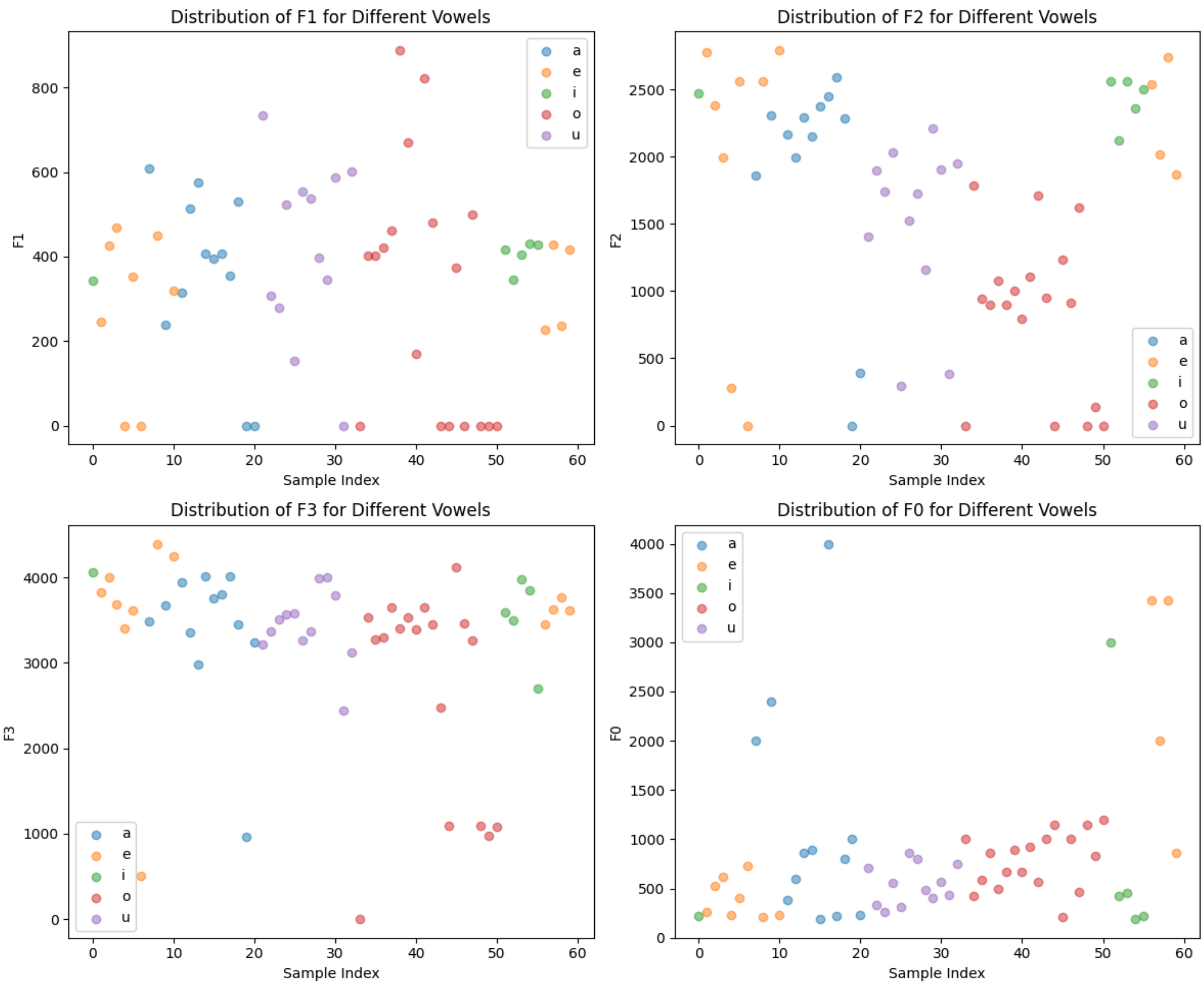
- A K-Nearest Neighbors (KNN) classifier with **k=5** and Euclidean distance was trained on the extracted features.

- The classifier was tested on unseen data, and predictions were evaluated against ground truth labels.

---

3. Results & Analysis

3.1 Visualization of Extracted Features



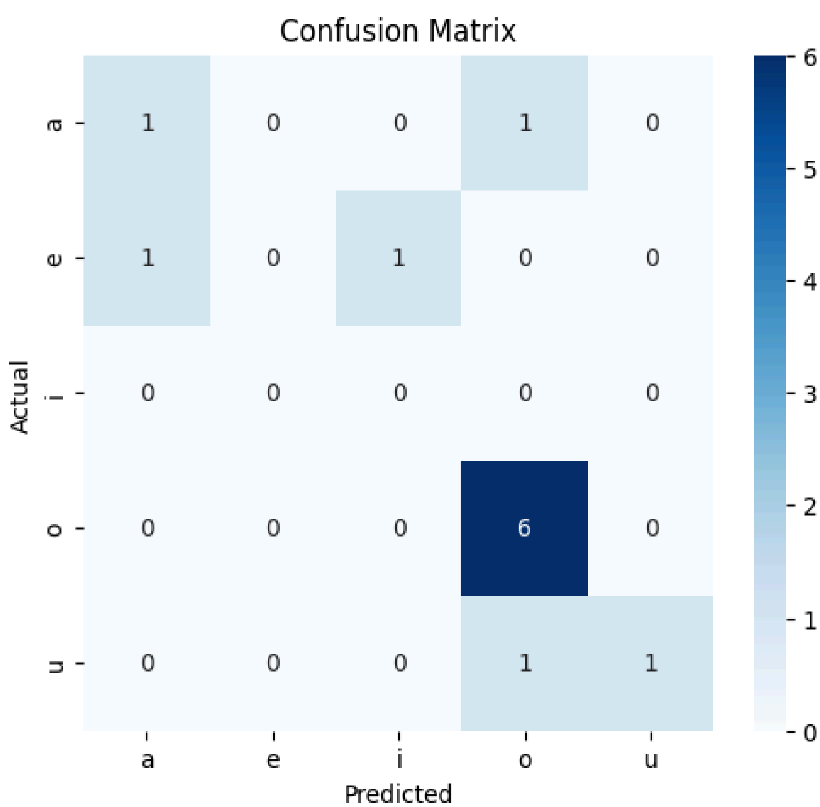
3.2 Classification Report

The KNN model achieved an overall accuracy of **66.67%**

Vowel	Precision	Recall	F1-score	Support
a	0.50	0.50	0.50	2
e	1.00	0.00	0.00	2
i	0.00	1.00	0.00	0
o	0.75	1.00	0.86	6
u	1.00	0.50	0.67	2

- **Overall Accuracy:** 66.67%
- **Macro Average:** Precision = 0.65, Recall = 0.60, F1-score = 0.40
- **Weighted Average:** Precision = 0.79, Recall = 0.67, F1-score = 0.62

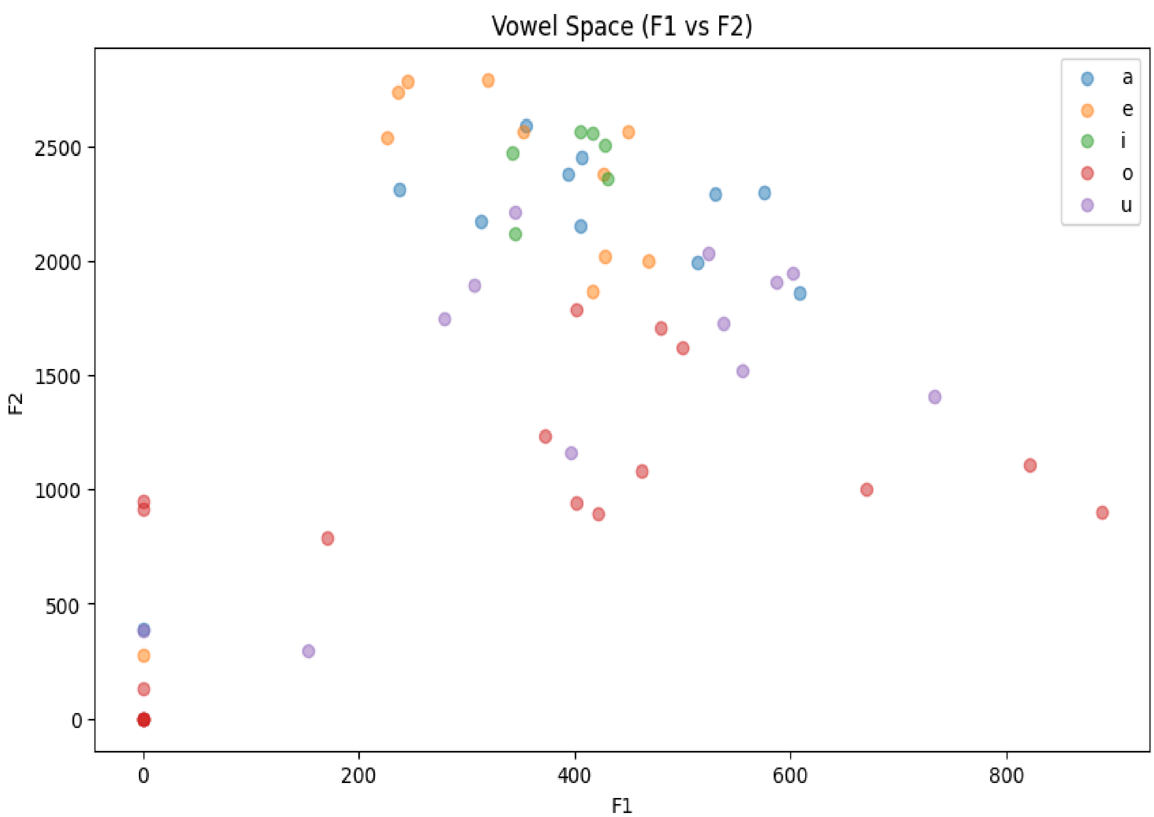
3.3 Confusion Matrix Analysis



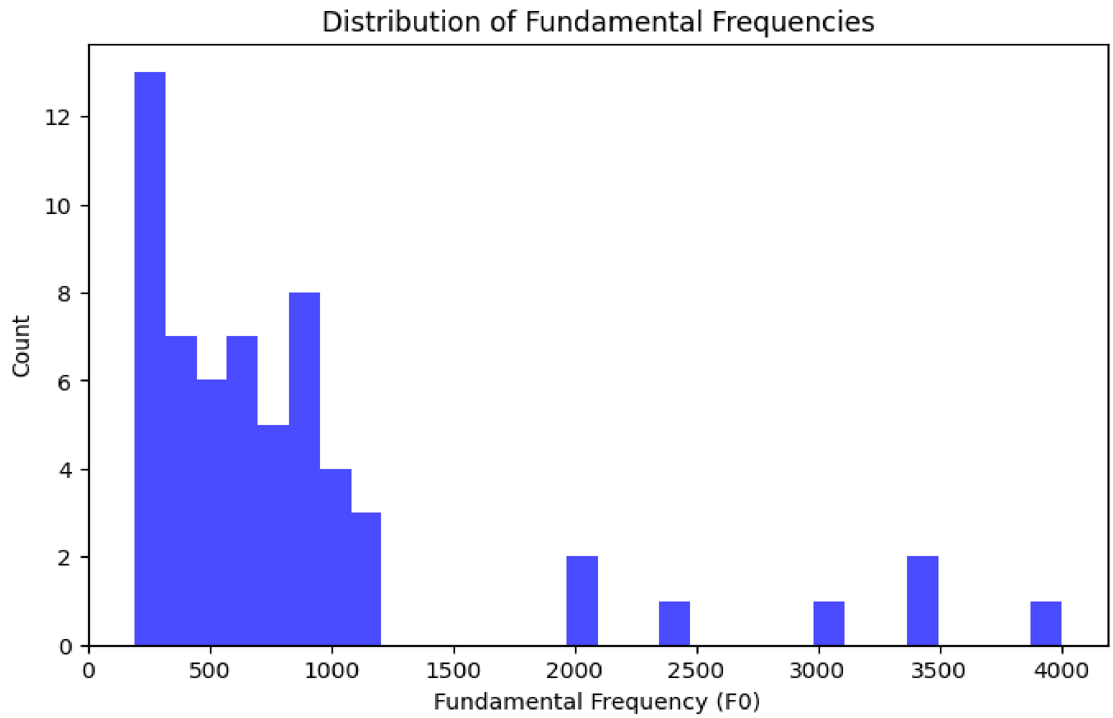
- **Strong performance** in classifying /o/ and /u/ vowels, with /o/ achieving 100% recall.
- **Misclassification observed** for /e/ and /i/, with /e/ not being correctly identified at all.
- **Confusion between certain vowel categories** suggests that feature overlap may still be a challenge.

3.4 Feature Distribution Insights

- The **vowel space plot (F1 vs. F2)** continues to show overlapping regions, making classification difficult.



- The **fundamental frequency (F0) distribution** suggests variability in speaker pitch, which could contribute to misclassification.



---

4. Discussion & Potential Improvements

4.1 Comparison with Theoretical Expectations

- **F1 correlates with vowel height** (low vowels have high F1, high vowels have low F1).
- **F2 correlates with vowel frontness** (front vowels have high F2, back vowels have low F2).
- **F3 relates to lip rounding and speaker characteristics.**
- **F0 reflects speaker pitch differences.**

The results align with phonetic theory but reveal errors due to dataset limitations and feature extraction inconsistencies.

4.2 Sources of Error

- **Overlapping Formant Values:** Certain vowels have similar formant frequencies, leading to classification difficulties.
- **Formant Estimation Issues:** LPC-based extraction is sensitive to noise, causing missing or incorrect values.
- **Feature Limitations:** Using only F1, F2, F3, and F0 may not fully capture vowel distinctions.
- **Dataset Imbalance & Speaker Variation:** Unequal sample sizes and speaker-dependent variations affect classification accuracy.

4.3 Relation to Historical Speech Recognition

- Early speech recognition systems relied on **formant tracking**, similar to this approach.
- Modern systems (e.g., **HMMs, DNNs**) use **MFCCs and probabilistic models** for improved robustness.
- This study reflects classical phoneme recognition techniques but could benefit from modern machine learning advancements.

4.4 Proposed Improvements

1. Improved Feature Engineering

- **Incorporate MFCCs** for a more robust spectral representation.
- **Use ΔF1, ΔF2, ΔF3** to track formant changes over time.
- **Energy and spectral entropy features** could further enhance classification.

2. Alternative Classification Methods

- **Support Vector Machines (SVMs):** Provides better separation compared to KNN.
- **Neural Networks (MLP, CNNs):** Can learn deeper vowel distinctions.
- **Gaussian Mixture Models (GMMs):** Probabilistic modeling could improve classification accuracy.
- **Hybrid Models:** Combining KNN with probabilistic models (e.g., Bayesian Networks) for more robust classification.

---

5. Conclusion

This demonstrates the feasibility of using **KNN for vowel classification** based on formant features. The **classification accuracy is 66.67%**, but misclassification remains an issue, particularly for vowels with overlapping formant values.

Key Takeaways:

- **Formant-based classification is effective** but has limitations due to feature overlap.
- **Additional acoustic features** (MFCCs, Δ-formants, spectral entropy) could enhance separation.
- **Alternative machine learning models** (SVM, Neural Networks, GMMs) should be explored for better performance.

Some improvements that could be made for better results focus on **hybrid approaches integrating spectral and formant-based features** for higher accuracy in vowel classification.

---

6. References

1. **os (Python Standard Library)**
   Python Software Foundation, "os — Miscellaneous operating system interfaces," Python Documentation, 2023. [Online]. Available: https://docs.python.org/3/library/os.html

2. **NumPy**
   C. R. Harris, K. J. Millman, S. J. van der Walt, et al., "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020. [Online]. Available: https://numpy.org/

3. **Librosa**
   M. McFee, B. McVicar, C. Raffel, et al., "librosa: Audio and Music Signal Processing in Python," Version 0.10.0, 2023. [Online]. Available: https://librosa.org/

4. **Matplotlib**
   J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90-95, 2007. [Online]. Available: https://matplotlib.org/

5. **SciPy**
   P. Virtanen, R. Gommers, T. E. Oliphant, et al., "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. [Online]. Available: https://scipy.org/

6. **Scikit-learn**
   F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/

7. **Seaborn**
   M. Waskom, "Seaborn: Statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021. [Online]. Available: https://seaborn.pydata.org/