

青　山　学　院　大　学

理　工　学　研　究　科

理 工 学 専 攻　知 能 情 報　コ ー ス

修　士　論　文

学 生 番 号　35623215

氏　　名　新 井　優 作

研究指導教員　ロ ペ ズ　ギ ョ ー ム 教 授

**Title:** Development of a Cooking Support System Using the Acoustic Characteristics of Fried Chicken

Student Name: Yusaku Arai

ID Number: 35623215

Degree: Master of Engineering

Course: Intelligence and Information

Thesis Advisor: Professor Guillaume Lopez

## Abstract

The recent spread of COVID-19 has increased opportunities to cook at home, and the number of novice cooks has increased. Fried foods present a significant challenge for novices because it is difficult to determine the proper frying conditions (heat, time, etc.). This research aims to develop a cooking support system, FCGS (Fried Chicken Goal system), which analyzes the cooking sound of fried chicken and judges the end of frying.

FCGS collects frying sounds using a smartphone microphone and detects the end of frying. It classifies the cooking sounds into two categories, “Middle” and “Finish,” and notifies the user in real-time when frying is finished. This makes it possible for even novice cooks to take out fried food at the appropriate time.

The following three types of sound features were investigated to analyze frying sound based on various research fields dealing with sound signal analysis.

- Mel-frequency cepstral coefficients (MFCC), which are widely used in voice recognition
- Wavelet coefficients, which are efficient in abnormality analysis from sound or vibration signal
- Amplitude spectrogram, which is used in music analysis and bioacoustics

In particular, the amplitude spectrogram can visualize frequency components that change with time, making capturing changes in fried sound easy. The study also introduces a change point detection method to determine the timing of the end of frying accurately.

Various machine-learning classifiers (SVM, k-NN, XGBoost) were tested to discriminate between mid-frying and end-of-frying. In addition, resampling techniques were used to deal with data imbalance problems. FCGS was evaluated by cross-validation using frying sound data. MFCC-based classification achieved an F1 score of 57.6%, and amplitude spectrograms enabled change-point detection, identifying the end of frying with 90% accuracy in F1 score, precision, and recall. In particular, the introduction of change-point detection significantly reduced the number of misjudgments during frying.

In this study, a cooking support system, FCGS, was developed to determine the end of frying by analyzing the sound of fried food cooking. Evaluation experiments demonstrated that the system could detect the end of frying accurately and effectively assist novice cooks. Future challenges include improving accuracy in different environments (noisy places), optimizing real-time processing, and applying machine learning methods.

# 理工学専攻修士論文要旨

提出年度 : 2024 年度  
提出日 : 2025 年 1 月 31 日  
専修コース : 知能情報コース  
学生番号 : 35623215  
学生氏名 : 新井 優作  
研究指導教員 : ロペズ ギヨーム 教授

(論文題目)  
唐揚げの音響特性を利用した調理支援システムの開発

## (内容の要旨)

近年、新型コロナウイルス感染症の拡大により、自宅で料理をする機会が増加し、料理初心者の増加が確認されている。特に揚げ物は、適切な加熱状態を見極めることが難しく、料理初心者にとって大きな課題となっている。また、加熱不十分な鶏肉は食中毒のリスクを伴うため、安全な調理が求められる。本研究では、唐揚げの調理音を分析し、揚げ終わりを判定する調理支援システム「FCGs (Fried Chicken Goal system)」を開発することを目的とする。

調理支援に関する研究として、音響センシングを活用した調理状態の認識が進められている。例えば、沸騰音や調理環境音を分析することで、調理プロセスの段階を特定する研究がある。また、食材の音響的特徴を利用し、成熟度や品質を判断する研究も存在する。本研究では、これらの知見を基に、唐揚げの調理音を用いた調理支援システムを構築する。FCGs は、スマートフォンのマイクを用いて唐揚げの調理音を収集し、機械学習によって揚げ終わりを検出するシステムである。理音を「揚げ途中 (Middle)」と「揚げ終わり (Finish)」の 2 つに分類し、リアルタイムで揚げ終わりをユーザーに通知する。これにより、料理初心者でも適切なタイミングで唐揚げを取り出すことが可能となる。

調理音の特徴量として、音波信号の解析が使われている様々な分野を参考に、次の 3 つを検討した。

- 音声認識分野に用いられているメル周波数ケプストラム係数 (MFCC)
- 音波・振動の異常検知分野に用いられているウェーブレット係数
- 音楽分析や生物音響学に用いられている振幅スペクトログラム

特に、振幅スペクトログラムは、時間とともに変化する周波数成分を視覚化でき、揚げ音の変化を捉えやすいことや、変化点検出手法を導入し、揚げ終わりのタイミングを高精度に判定することが期待できる。

識別手法として、複数の機械学習分類器 (SVM, k-NN, XGBoost など) を用いて揚げ途中と揚げ終わりの判別を行う。さらに、リサンプリング手法を活用し、データの不均衡問題に対応する。また、変化点検出手法を用いることで、調理中の音の変化を解析し、適切なタイミングで揚げ終わりを判定する。FCGs の評価は、調理音データを用いた 1 データ抜き交差検証によって行われた。MFCC を用いた分類では、F1 スコアが 57.6% を達成し、振幅スペクトログラムを基にした変化点検出では、揚げ終わりの検出精度が F1 スコア、適合率、再現率の 3 指標において 90% を達成した。特に、変化点検出を導入することで、揚げ途中での誤判定を大幅に削減できた。

本研究では、唐揚げの調理音を分析し、揚げ終わりを判定する調理支援システム FCGs を開発した。評価実験の結果、本システムは高い精度で揚げ終わりを検出可能であり、料理初心者の調理を支援する効果が期待される。今後の課題として、異なる環境（騒音がある場所）での精度向上、リアルタイム処理の最適化、さらなる機械学習手法の適用が挙げられる。また、スマートスピーカとの連携やマルチモーダル解析（画像+音）を活用することで、より高度な調理支援システムの開発が可能となる。

唐揚げの音響特性を利用した  
調理支援システムの開発

新井 優作

2025/01/31

# 目 次

<b>第 1 章 序論</b>	<b>3</b>
1.1 研究背景 . . . . .	3
1.1.1 調理支援システムの需要 . . . . .	3
1.1.2 センシング技術を利用した調理支援 . . . . .	4
1.1.3 揚げ物の揚がり具合判断 . . . . .	5
1.1.4 加熱不十分な鶏肉調理の危険性 . . . . .	6
1.1.5 音の検出・認知を行う既存製品 . . . . .	6
1.2 研究目的 . . . . .	7
1.3 本論文の構成 . . . . .	8
<b>第 2 章 調理支援および音響的特徴を利用した関連研究</b>	<b>9</b>
2.1 音響センシングによる調理支援に関する研究 . . . . .	9
2.2 食材の音響的特徴を利用した状態認識に関する研究 . . . . .	11
2.3 異常音検出に関する関連研究 . . . . .	13
2.4 画像認識による調理支援および食材の状態認識に関する研究 . . . . .	15
<b>第 3 章 FCGs: 料理初心者に向けた唐揚げ調理支援システム</b>	<b>17</b>
3.1 FCGs の提案 . . . . .	17
3.1.1 唐揚げの調理状態の定義 . . . . .	17
3.1.2 FCGs のシステム構成 . . . . .	17
<b>第 4 章 調理音の特徴抽出手法および識別手法</b>	<b>19</b>
4.1 調理音および特微量の解析手法 . . . . .	19
4.1.1 スペクトログラム解析 . . . . .	19
4.1.2 主成分分析 . . . . .	20
4.1.3 t 分布型確率的近傍埋め込み法 . . . . .	20
4.2 調理音から抽出される音響特微量 . . . . .	21
4.2.1 振幅スペクトログラム . . . . .	21
4.2.2 MFCC . . . . .	21
4.2.3 ウェーブレット変換 . . . . .	23
4.3 調理段階の識別手法 . . . . .	24
4.3.1 機械学習分類手法 . . . . .	24
4.3.2 リサンプリングによる不均等データ処理手法 . . . . .	27
4.3.3 調理音の変化点検出手法 . . . . .	29
4.3.4 調理音に含まれる雑音の処理 . . . . .	30
4.4 各識別手法の評価方法および評価指標 . . . . .	31

4.4.1	一グループ抜き交差検証による評価方法	31
4.4.2	評価指標	31
<b>第5章</b>	<b>調理音識別手法およびFCGsの開発</b>	<b>33</b>
5.1	調理音データのラベリング	33
5.2	機械学習モデルによる唐揚げの調理音データ分類精度の評価	33
5.2.1	MFCC特徴量を使用した分類精度の評価	34
5.2.2	PCAおよびt-SNEによるMFCC特徴量解析	37
5.2.3	リサンプリングバランスの調整による分類モデルの再評価	38
5.2.4	振幅スペクトログラムに基づく再ラベリング	39
5.2.5	4クラスにおけるMFCC特徴量による分類精度の評価	39
5.2.6	ウェーブレット変換による特徴量抽出を利用した分類精度の評価	42
5.3	変化点検出による唐揚げの揚げ終わり検出精度の評価	42
5.4	変化点検出を利用した揚げ終わり判定システムの実装	46
<b>第6章</b>	<b>結論</b>	<b>52</b>
6.1	本研究のまとめ	52
6.2	今後の展望	52
<b>謝辞</b>		<b>54</b>
<b>参考文献</b>		<b>55</b>
<b>付録A</b>	<b>本研究に関する論文と発表実績</b>	<b>62</b>
<b>付録B</b>	<b>唐揚げの一般的なレシピまとめ</b>	<b>63</b>
<b>付録C</b>	<b>MFCC特徴量を使用したその他の分類精度の評価</b>	<b>64</b>
C.1	アンダーサンプリングにCluster Centroidを使用した際の分類精度一覧	64
C.2	アンダーサンプリングにCondensed Nearest Neighbourを使用した際の分類精度一覧	66
C.3	アンダーサンプリングにNearMiss-1を使用した際の分類精度一覧	67
C.4	アンダーサンプリングにNearMiss-2を使用した際の分類精度一覧	68
C.5	アンダーサンプリングにOne-Sided Selectionを使用した際の分類精度一覧	69

# 第1章 序論

本章では序論として、本研究における背景および研究目的、本論文の構成について述べる。

## 1.1 研究背景

### 1.1.1 調理支援システムの需要

#### 料理初心者の増加

新型コロナウイルス感染症の拡大以前は、フードデリバリーサービスが普及したこともあり、外食・中食の利用頻度が増加の一途を辿っていた[1][2]。反対に、家庭で料理をする機会は減少していた。しかし、新型コロナウイルス感染症の拡大により、在宅時間が増えると共に、料理をする機会も増加した。

2021年に実施された農林水産省による「食育に関する意識調査」では、図1.1で示されているように「自宅で食事を食べる回数」・「自宅で料理を作る回数」が増えたと回答した割合がそれぞれ35.5%，26.5%となった。特に若い世代（20～39歳）において、「自宅で料理を作る回数」が増えたと回答した割合は39.5%にまで上った[3]。2020年の消費者動向調査においても、コロナ禍での調理時間・回数が増えたと回答した割合は全体で32.9%となった[4]。

この傾向は海外でも見られる。スペイン・クロアチア・ニュージーランドでそれぞれ実施された調査では、新型コロナウイルス感染症の流行によるロックダウン中に調理頻度が増加したもの割合は約4～5割という結果が報告されている[5][6][7]。

このように、新型コロナウイルス感染症が拡大した結果、料理をする機会が増加している。これに伴い、普段料理をする機会がなかった人々が料理をする機会を得たため、料理初心者が増加していると考えられる。

#### 調理技術による料理の壁

料理初心者の大きな壁として、レシピ表現の曖昧さおよび調理技術の不足が挙げられる。

レシピ表現の曖昧さに関しては調味料の量をあらわす「少々」および「適量」、火加減における食材の色・状態変化の見極めが壁となっている。ハウス食品株式会社が男女500人ずつの計1000人を対象に実施した調査によると「レシピの独特的な表現に戸惑ったことがある」と回答した人は約6割である。そのうち、コロナ禍で料理を始めた人は約8割に上る[8]。

調理技術においては食材を適切な大きさに切るといった基本的な技術から、魚の三枚おろしをはじめとした多様な包丁さばきなどが料理初心者にとって習得に時間のかかる技術であると考えられる。平島らの調査によると、半月切りや輪切りという基本的な包丁の技

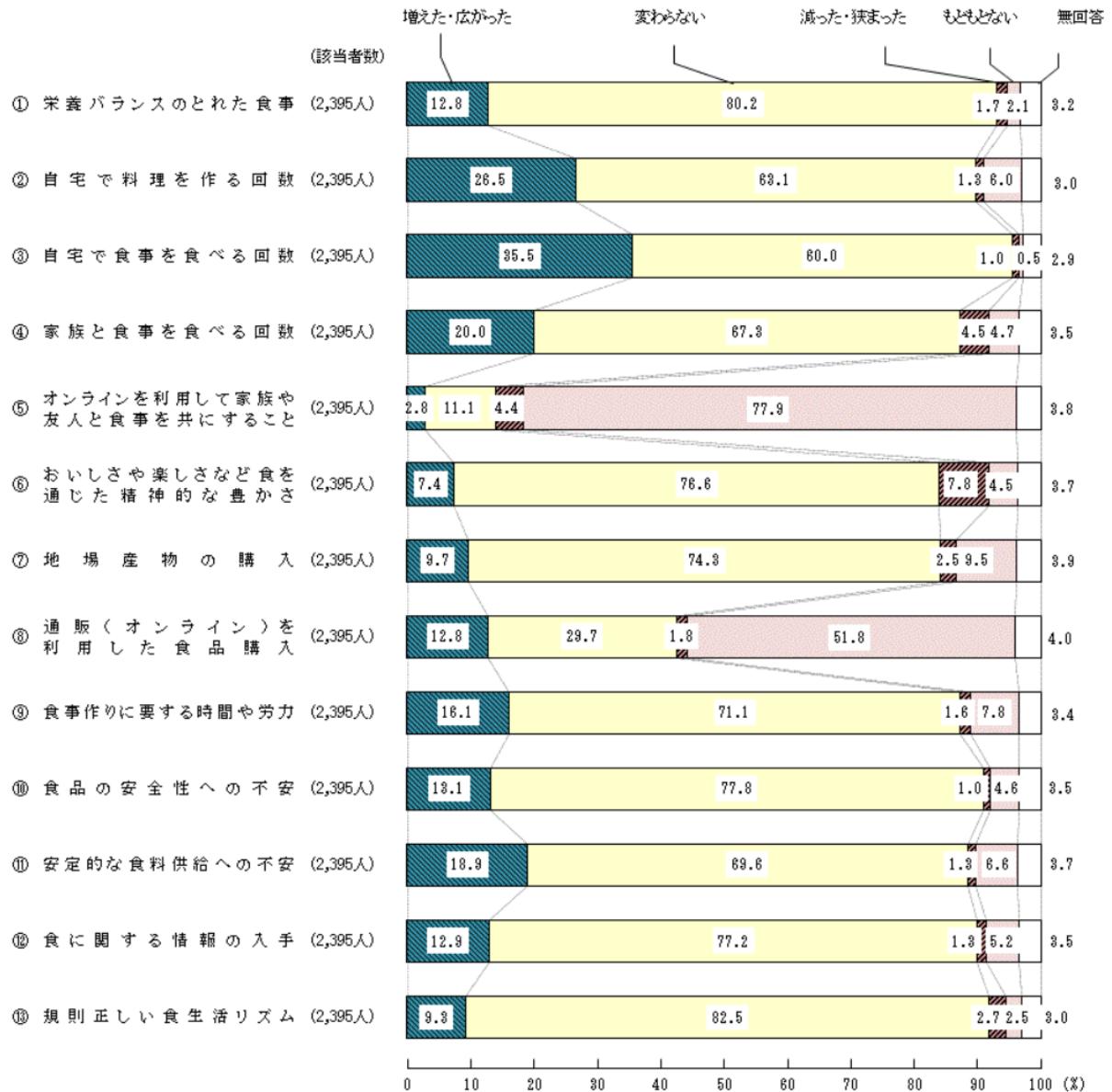


図 1.1: 新型コロナウイルス感染症による食生活の変化 ([3] より引用)

術において、普段調理をほとんどしない 235 人のうち、それぞれ、18.7% と 8.9% の人ができないと回答している [9].

よって、料理初心者がすぐに習得することが困難である技術および調理における判断を支援するシステムが必要であると考えられる.

### 1.1.2 センシング技術を利用した調理支援

料理初心者を支援するための技術として、センシング技術が挙げられる. 近年、IoT (Internet of Things) 技術の普及によりスマートフォンおよびタブレットだけなく、日常のあらゆるものにセンサが搭載されている. 調理器具およびキッチン用品もその例外ではない. 調理器具およびキッチン用品とセンシング技術や IoT 技術を導入したスマートキッチン家電は、料理初心者の未熟な技術および判断を支援し、調理の負担を軽減する. Kido らの研究では、調味料の容器にモーションセンサ、LED ライトを用いることで調味料追加

の支援システムを提案した[10]. 加藤らの研究では、包丁の柄先に加速度センサを用いるレシピ動画の自動同期システム SynCook を提案した[11]. また、2021 年には Hestan Cue という調理器具、IH ヒーター、ビデオガイド付きレシピを組み合わせたスマート調理家電が発売された[12]. 図 1.2 のように調理器具、IH ヒーター、スマートデバイスが Bluetooth で連動し、調理器具に組み込まれている温度センサにより最適な温度をリアルタイムで検知、加熱温度をアプリを介して IH ヒーターに指示する. さらに調理器具に入れるタイミングおよびひっくり返すタイミング、取り出すタイミングなどを通知する.

このように料理をする際に利用するものとセンシング技術の融合により、日々の料理が可制御、可視化されていることがわかる.



図 1.2: スマート調理家電「Hestan Cue」のシステム概要 ([12] より引用)

### 1.1.3 揚げ物の揚がり具合判断

揚げ物は食品に小麦粉・溶き卵・パン粉・片栗粉などをコーティングし、高温の油で揚げることで表面の水分を蒸発させサクサクの食感を実現している. 周りの衣および食品のサイズ、揚げる時間によって揚げ物の食感も変わる[13]. そのため衣は色が付き十分揚がっているように見えるが中身に火が通っていない場合がある. このように、表面だけで判断することが難しい揚げ物は、料理初心者にとっては困難なレシピの 1 つであると考えられる.

そこで本研究では揚げ物に着目する. 揚げ物を揚げる際、衣の水分損失が指数的に減少することから、揚げ物の揚がり具合による水分と油の振動が音響的変化にあらわれると想定される[14]. よって、揚げ物を揚げている時間に対する音響変化を利用した、揚がり具合の判定が可能であると考えられる.

#### 1.1.4 加熱不十分な鶏肉調理の危険性

揚げ物料理の中でも、加熱不足で食中毒となる危険性が高い食材は鶏肉である。生の鶏肉はサルモネラやカンピロバクター等の食中毒菌の感染源となる[15][16]。特に、カンピロバクター食中毒はノロウイルスと並んで食中毒事件数が高い値となっている。厚生労働省が実施した調査では、令和5年においてカンピロバクター食中毒が発生した件数が211件に上っていることを報告している[17]。結果、日本では各機関が予防のために注意喚起している[17][18][19]。食中毒を防ぐための十分な加熱は、鶏肉の中心温度が75°C以上で1分間以上とされている。

しかし Solveig らの調査によると、家庭での鶏肉調理の大半が内部温度を測定せずに調理されており、病原体の完全な無害化を達成できていない場合が多いことがわかった[20]。

よって本研究では、加熱不十分な状態を調理完了と誤判定しない安全な調理支援システムを構築する。

#### 1.1.5 音の検出・認知を行う既存製品

今日では、様々なスマートデバイスが普及したことにより、音を利用する技術が日常生活に馴染んでいる。本項では、既に一般に普及している音を利用した技術を紹介する。

##### 音声センサによる音検知通知

Google および Apple はスマートデバイスが周囲の物音を聞き取り通知してくれる機能を提供している[21][22]。本機能は聴覚の障害がある人およびイヤホン・耳栓を装着しているとき、物音に気付きにくい状況下でも、指定の音を検知すると図1.3のようにスマートデバイスへ直接通知をしたり、端末の振動およびカメラ撮影用ライトを点滅させたりして通知をしてくれる。このようにスマートデバイスに内蔵されている音声センサのみで音の検知が可能になっている。

また、スマートフォンの内臓マイクロフォンなどの音響的センサは身近な機械に組み込むことが容易であり、料理中に水がかかることおよびセンサの設置が不自由であるといった課題が解消される。

##### スマートスピーカの普及

Apple が自社のスマートフォンに搭載した「Siri」および Amazon が発売した「Amazon Echo」などを皮切りにスマートスピーカーの市場は成長を続けている。野村総合研究所の「IT ナビゲーター 2021 年度版」[23] によると 2020 年のスマートスピーカ世帯普及率は 11.2% (583 万世帯) だったが、2021 年には 13.5% (691 万世帯) と増加している(図1.4)。さらに 2026 年には 30.8% (1544 万世帯) にまで上ることが見込まれている。

このような現状から、本研究では音を利用した料理支援に着目する。

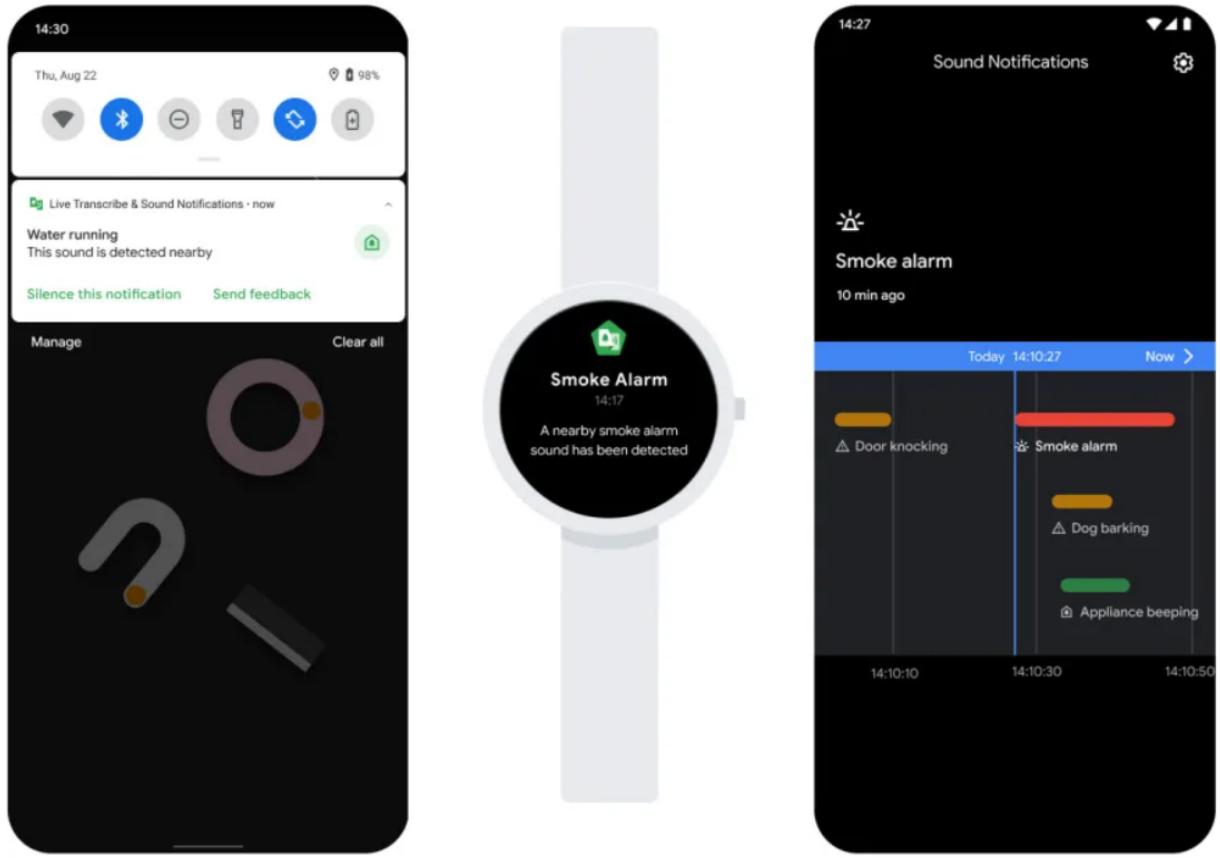


図 1.3: Android 端末の音検知通知の様子 ([21] より引用)

## 1.2 研究目的

前節で述べたように、新型コロナウイルス感染症拡大の影響から現代社会の料理に対する向き合い方に変化が生じており、料理初心者に対する料理支援の必要性が増加している。また、揚げ物料理に対する料理初心者の壁として、火加減の調整および中身まで火が通っているかの判断などの難しさが存在する。音の技術に関しては、既に身近にある様々な製品にその技術が組み込まれているため馴染みのある技術だと言える。以上の点から本論文では身近な端末であるスマートフォンに内臓されているマイクロフォンから音響信号を取得し、揚げ物の中でも特に中身が十分加熱されているかわかりづらく、加熱不十分による食中毒の危険性を持つ鶏肉料理の一つである唐揚げを対象とする。そして、唐揚げの揚げ終わりを判別する調理支援システムの開発を目的とする。

そこで本研究では次の 2 つを目標とする。

1. 唐揚げの調理状態において、揚げ終わりとなる前の状態で調理完了と誤判定しない安全な音判定システムの構築。
2. スマートフォン上で唐揚げの揚げ終わりをリアルタイムに判別する、容易に利用可能なアプリケーションの開発。

調理音から振幅の平均を算出し、変化点検出を実施した結果、揚げ終わりの検出は F1 スコア、適合率、再現率の 3 種類の指標で 90% の精度を達成した。また、変化点検出アル

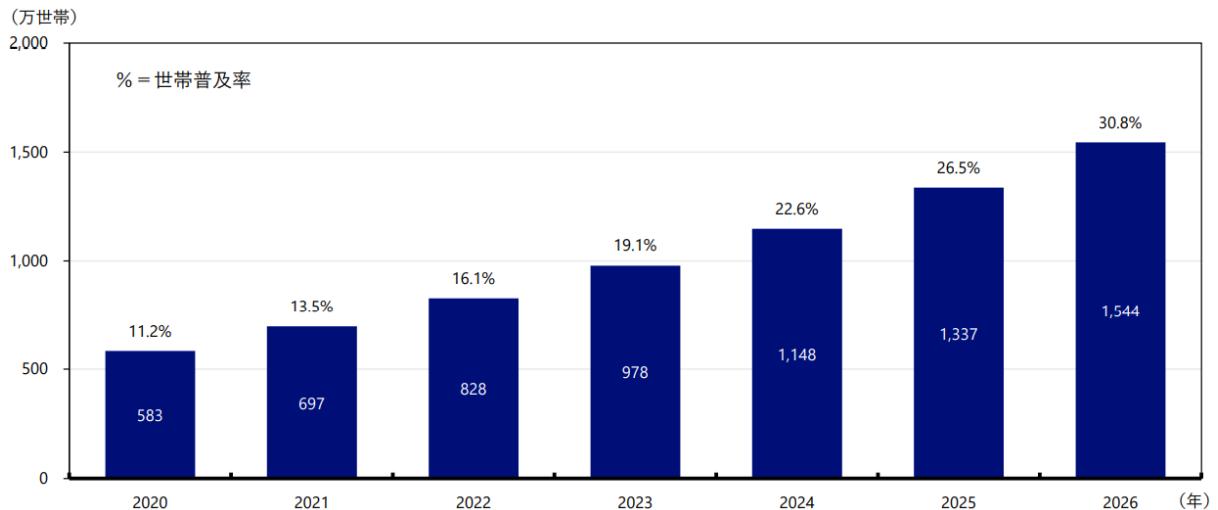


図 1.4: 日本におけるスマートスピーカの保有世帯数・普及率予測 ([23] より引用)

ゴリズムを実装した、リアルタイムで唐揚げの調理完了タイミングを判定するスマートフォンアプリケーション FCGs を開発した。

### 1.3 本論文の構成

本論文の構成を以下に示す。

#### 第 1 章: 序論

本論文の研究背景、研究目的および論文の構成について述べる。

#### 第 2 章: 調理支援および音響特徴を利用した関連研究

本研究に関連する、調理支援に関する研究および音響特徴を利用した研究について述べる。

#### 第 3 章: FCGs: 料理初心者に向けた唐揚げ調理支援システム

唐揚げの調理支援システムの提案について述べる。

#### 第 4 章: 調理音の特徴抽出手法および識別手法

本研究で検討した唐揚げの調理音から得られる音響特徴量および調理完了の識別手法について述べる。

#### 第 5 章: 調理音識別手法および FCGs の開発

各識別手法の評価および調理支援システムの評価について述べる。また、提案手法の開発について述べる。

#### 第 6 章: 結論

本論文の結論と今後の展望について述べる。

# 第2章 調理支援および音響的特徴を利用した関連研究

本章では本研究を進めるにあたっての関連した研究についてまとめる。

## 2.1 音響センシングによる調理支援に関する研究

調理中の動作および食材を音で認識することで、調理支援が可能である。そのため、調理活動認識および食材の状態認識に関する研究が実施されている。

Tabacchi らは、調理プロセスの段階を音響および振動データに基づいて分類する新しい統計的パターン認識手法を提案した[24]。本研究では、水の沸騰という単純なケースを対象とし、調理段階の識別精度を向上させる最適化モジュールを導入した。

沸騰段階のデータは、鍋に水を 1.5 L 入れ、室温から最大加熱力で加熱することで収録された。記録には、AKG SE300B カーディオイドマイク (44.1 kHz) と PCB 加速度計 (352C33, 44.1 kHz) が使用され、合計 28 回の実験が実施された。また、録音環境は静かな部屋で統一された。特微量として MFCC (Mel-Frequency Cepstral Coefficient) が使用され、加熱、核沸騰、遷移沸騰、膜沸騰の 4 つの段階が識別された。各録音は 100 ms ごとに分割され、Parzen Classifier を用いて分類が行われた。音響データ単独、振動データ単独、および両者の組み合わせの 3 種類の特徴セットが試験された。

最適化モジュールでは、分類結果に移動平均フィルタを適用し、閾値を超える確率が 1 秒以上続く場合にクラスが確定される仕組みが導入された。この結果、加熱および膜沸騰段階では最適化後に 100% の識別精度が達成された。一方、核沸騰および遷移沸騰段階では、最適化前の識別精度は約 98% であったが、最適化により誤分類が減少した。

今後の課題として、振動データのラベル付けが音響信号に基づいて実施されたことによる精度低下が指摘されている。振動データの独自のラベリング手法の導入や、温度センサや圧力センサなど他のデータと統合したマルチモーダル処理が求められる。

宮澤らは、調理環境音を用いた行動認識においてレシピ情報を時系列的に考慮し、音響識別器の中間特微量を導入することで認識精度の向上を図った[25]。従来の手法では「切る」「焼く」といった調理行動のみを対象としていたが、本研究では「水の音」や「ビニール袋の擦れる音」などの環境音も認識対象に加えることで、行動識別の精度がさらに向上することが示された。

焼きそばの調理を対象とし、音響データは iPhone 8 で録音された 6 つの調理環境音を使用した。録音された音声データは、サンプリング周波数 16 kHz で記録された。特微量として MFCC と、音響識別器の中間層から抽出した特微量が使用された。音響識別器には EnvNet-v2 を改良したモデルが採用され、ボトルネック層の出力を新たな特微量として利用することで、調理行動の識別精度向上が図られた。

調理行動の識別には HMM (Hidden Markov Model) が使用され、モデルの出力確率分布は GMM (Gaussian mixture models) で推定された。実験では、MFCC の窓幅を 16 ms, 160 ms, 1600 ms の 3 条件で設定し、HMM の混合数 1・2 の条件で識別精度が評価された。評価指標として F 値が用いられ、cut (切る)・grill (焼く) の調理行動に加え others (その他) が対象となった。HMM ネットワークは図 2.1 のようになる。

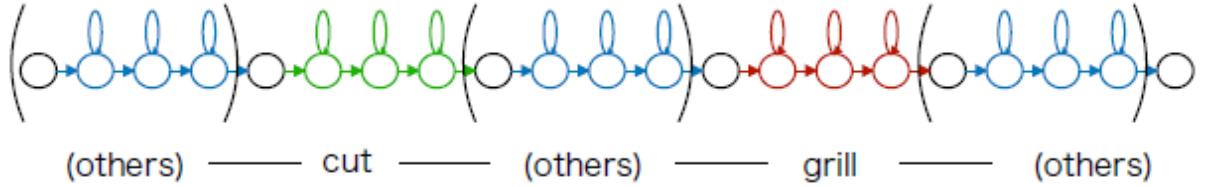


図 2.1: レシピ情報を加味した HMM ネットワーク ([25] より引用)

結果として、MFCC と比較して音響識別器の中間特徴量を導入した方が「焼く」行動の識別精度が向上した。特に、中間特徴量の次元を 100 に設定した場合、F 値が最も高くなり、音響識別器を用いる有効性が示された。一方、「切る」行動に関しては中間特徴量の導入による精度向上は確認されなかったものの、全体的な識別精度は向上した。さらに、「その他」クラスを「水の音」「ビニール袋の音」などに細分化することで、他クラスとの誤認識が減少し、識別性能が向上することが確認された。

今後の課題として、GMM の混合数の増加が必ずしも識別精度向上に寄与しないことが指摘された。特に「切る」音と環境音 (others) が類似している場合、誤分類の原因となる可能性がある。環境音クラスをさらに細分化することで、識別精度を高める必要がある。また、実験はオフラインで実施されたが、オンラインのリアルタイム認識システムへの応用が求められる。

陳らは、音響解析を用いて唐揚げの揚がり具合を判定する手法を提案した [26]。本研究では、スマートフォンの内蔵マイクを活用して調理中の音響データを取得し、機械学習を用いて唐揚げの調理状態を分類することを目的とした。従来の調理支援システムは、調理器具にセンサを取り付ける必要があったが、本手法では外部機器を必要とせず、より簡便に導入可能なシステムが設計された。

唐揚げの調理音を対象とし、データは iPhone 11 を使用して静かなキッチン環境で録音された。サンプリング周波数は 44.1 kHz で、調理油の温度はクックサーモ 5495B を使用して監視された。表 2.1 の通り、温度 (150°C, 170°C, 200°C), 個数 (5 個, 7 個, 10 個), 調理油の使用回数 (1 回目, 2 回目) の条件を変えて 10 回分のデータが収集された。調理音は「揚げはじめ」「変化 1」「変化 2」「変化 3」「揚げ上がり」の 5 つのラベルに分類され、音響データには MFCC が特徴量として使用された。

機械学習には、自動機械学習ツール「TPOT」を使用した。TPOT は遺伝的プログラミングにより最適なモデルを自動探索し、ハイパーパラメータの調整を実施する。本研究では、5 分割交差検証を用いて複数のモデルを試行した結果、XGB (eXtreme Gradient Boosting) が最も高い精度を達成した。

実験の結果は表 2.2 の通り、全体の F1 スコアは 97.14%，揚げ上がり段階の F1 スコアは 85.71% に達した。揚げはじめや変化段階は 95% 以上の精度で分類されたが、揚げ上

表 2.1: 計測 10 回分の唐揚げの揚げ方 ([26] を基に作成)

	温度 [°C]	個数 [個]	油使用回数 [回]
動画 1	170	7	1
動画 2	170	7	2
動画 3	170	10	1
動画 4	170	10	2
動画 5	170	5	1
動画 6	170	5	2
動画 7	150	7	1
動画 8	150	7	2
動画 9	200	7	1
動画 10	200	7	2

り段階の精度は他のクラスよりも低かった。

表 2.2: 各クラスにおいての評価結果 ([26] より引用)

評価尺度	その他	揚げはじめ	変化 1	変化 2	変化 3	揚げ上がり	平均	加重平均
Recall	98.53%	95.71%	98.66%	95.96%	98.58%	83.33%	95.13%	97.15%
Specificity	99.79%	99.63%	98.14%	99.52%	99.40%	99.80%	99.38%	99.15%
Precision	97.10%	98.67%	95.45%	98.17%	97.66%	88.24%	95.88%	97.17%
Accuracy	98.53%	95.71%	98.66%	95.96%	98.58%	83.33%	95.13%	97.15%
F1-score	97.81%	97.17%	97.03%	97.05%	98.12%	85.71%	95.48%	97.14%

揚げ上がり段階の精度が低下した要因として、データ数の不足が挙げられる。特に高温や低温、量が多い状態では音響特性が異なり、単純な閾値設定では分類が困難であった。しかし、TPOT による機械学習モデルの最適化により、温度や個数の条件が異なっても一定の精度が維持された。

本研究は、音響解析と機械学習を組み合わせて調理支援を行う新たなアプローチを提案しており、外部センサを必要とせず、スマートフォンだけで調理状態を判定できる点が特徴である。今後は、より多くの調理データを収集し、揚げ上がり段階の精度向上を目指すとともに、リアルタイムでの揚がり具合判定システムの構築が求められる。

## 2.2 食材の音響的特徴を利用した状態認識に関する研究

食材の状態を正確に把握することは、食品の品質管理および食材の美味しさの認識に有効である。そのため、非破壊的な食材の状態認識および食感の分類に関する研究が実施されている。

Caladcad らは、音響信号に基づいてココナッツの成熟度を判定する手法を提案した [27]。本研究では、ココナッツを叩くことで得られる音響データを取得し、機械学習を活用して成熟度を分類することを目的としている。従来の成熟度判定は農家が音を頼りに行う主観的な方法であったが、本手法では音響データを解析することで、より客観的で高精度な分類を目指している。

ココナッツのタッピング音は、専用のタッピングチャンバを用いて収録された。サンプリング周波数は 44.1 kHz で、吸音材を用いたチャンバ内でノイズを抑えながら録音された。収録された音響データは、高速フーリエ変換 (Fast Fourier Transform, FFT) を施し、周波数領域の特徴量を抽出した。サンプル数は 129 個（未成熟 8 個、成熟 36 個、過成熟 85 個）で、それぞれ 3 回ずつタッピングされ、合計 387 のデータが得られた。

機械学習モデルとして、人工ニューラルネットワーク (Artificial Neural Network, ANN), サポートベクターマシン (Support Vector Machine, SVM), ランダムフォレストの 3 種類が用いられた。データは 70% を訓練用、30% をテスト用として分割し、10 分割交差検証によりモデルの評価が実施された。

実験の結果は表 2.3 の通りである。ランダムフォレストが最も高い精度を達成し、訓練精度は 90.98%，テスト精度は 83.48% に達した。一方、ANN は訓練精度 79.32%，テスト精度 81.74% で安定した結果を示し、SVM はやや精度が低かった。特に過成熟の分類精度は高かったが、未成熟と成熟の識別はやや難しく、データの不均衡が影響したと考えられる。

表 2.3: 3 種類の機械学習分類器の性能比較 ([27] より引用)

Model	Classification Accuracy [%]		F-Score [%]	
	Train	Test	Train	Test
Artificial Neural Network	79.32	81.74	77.46	79.27
<b>Random Forest</b>	<b>90.98</b>	<b>83.48</b>	<b>91.41</b>	<b>81.35</b>
Support Vector Machine	88.35	80.00	88.79	76.67

原因として、未成熟のサンプル数が少ないことが分類精度の低下を引き起こしている可能性が指摘された。今後は、サンプル数の増加やデータ拡張を実施し、未成熟クラスの精度向上を図る必要がある。また、深層学習モデルの導入による分類精度向上や、リアルタイム判定が可能なシステムの構築が求められる。

Zhao らは、芯にかびを持つ不健全なリンゴを非破壊で検出するために、振動音響信号の多領域特徴を用いた深層学習と浅層学習を組み合わせたハイブリッドモデルを提案した [28]。本研究は、健康な果実、軽度のかび芯（不健全）、重度のかび芯（病果）の 3 クラス分類を目指し、果実内部の初期異常を検出する方法を開発している。

かび芯はリンゴの内部障害であり、初期段階では外観上の異常がなく、食品安全上のリスクが見過ごされやすい。本研究では、リンゴの赤道帯部分に電圧刺激を加え、誘発された振動音響信号を圧電センサを用いて収集した。振動音響信号は時間領域、周波数領域、時間-周波数領域の 2 次元画像に変換され、多領域特徴の抽出に利用された。ResNet50 で深層特徴を得た後、SVM および Extreme Learning Machine (ELM) の浅層学習モデルで分類するハイブリッド構造を採用し、特徴選択と分類器の最適化には粒子群最適化 (Particle Swarm Optimization, PSO) アルゴリズムが用いられた。

リンゴ 1200 個を対象とした実験では、時間-周波数領域の特徴が特に効果的であり、提案モデル (IResNet50-PSO-ELM) は全体で 96.7% の分類精度を達成した。健康な果実、不健康な果実、病果の分類精度はそれぞれ 100%，94.1%，96.2% であり、不健康な果実の一部が健康な果実または病果と誤分類される課題が残ったものの、F1 スコアは全クラスで

95%以上を記録した。

本研究は、果実内部の初期障害を高精度で非破壊検出する可能性を示しており、振動音響データを利用した食品品質管理の有用性を提案している。

Lopez らは、食品のサクサク感を音響信号に基づいて分類する手法を提案した[29]。本研究では、Autonomous Sensory Meridian Response (ASMR) 動画から音声データを収集し、フライドチキン、ポテトチップス、トーストのサクサク感を深層ニューラルネットワーク (Deep Neural Network, DNN) で分類することを目的としている。従来の官能評価や機械的測定と異なり、音響データを用いることで、非侵襲的かつ迅速に食品の食感を評価することが可能となる。

本研究では、YouTube の ASMR 動画 584 本からフライドチキン、ポテトチップス、トーストの咀嚼音を収集した。音声データは 1 秒ごとにトリミングされ、サンプリング周波数を 22,050 Hz に統一して保存された。収集した音声から、MFCC および離散フーリエ変換 (Discrete Fourier Transform, DFT) スペクトルが特徴量として抽出された。また、テストデータとして音響防音ボックス内で機械的に圧縮して収録されたフレッシュトースト、ASMR 動画から収集されたミルクに浸したトーストの音響データを用意した。

分類モデルには、MFCC を用いた多層パーセプトロン (Multi-layer Perceptron, MLP) と、DFT スペクトルを入力とする ResNet (残差ネットワーク) が使用された。MLP は 512, 256, 3 のニューロンを持つ 3 層の全結合ネットワークで構成され、ResNet は複数の残差ブロックからなる深層学習モデルである。

結果、MLP モデルは学習データで 95%，テストデータで 100% の精度を達成した。一方、ResNet モデルは学習データで 85%，データ拡張後には 97% の精度を示したが、テストデータでは 16% と低い精度にとどまった。特に、MLP モデルはフレッシュトーストとミルクに浸したトーストの音響データも高精度で分類し、外部データに対する汎化性能が示された。

本研究は、食品のサクサク感を音響信号から分類する新たな手法を提示し、食品業界における品質管理や製品開発に貢献する知見を提供している。今後は、さらなるデータセットの拡充やリアルタイムでの分類システム構築が求められる。

本研究においても、唐揚げのフライ調理音に最適な特徴量および分類器を利用することで、揚げ終わりの特徴を捉え、正確な識別が可能であると考える。

## 2.3 異常音検出に関する関連研究

異常音検出は、注意を向けてないと気づくことが困難な変化を認識可能にする。そのため、わずかな音の変化を捉えるための研究が実施されている。

Xiao らは、ガスピライインの漏れを音響信号に基づいて検出する手法を提案した[30]。本研究では、ウェーブレット変換と SVM を組み合わせた漏れ検出および深刻度分類システムを開発し、パイプラインの安全性を向上させる技術について検討している。特に、漏れ音の特性を利用し、高精度かつ迅速に漏れの規模の識別が可能であることが示されている。音響センサによって収集された漏れ音はウェーブレット変換によって特徴抽出が行われ、ノイズ除去後に SVM で分類される。

本システムの開発には、30 m および 3 m のパイプラインが使用され、直径 1 mm, 3

mm, 5 mm の漏れ穴を設けた実験が実施された. 結果として, 漏れの有無は 99.4% の精度で検出され, 漏れの深刻度は 95.6% の精度で分類された. 特に, 標準偏差, ウエーブレット平均周波数, および絶対平均が漏れの深刻度分類において重要な役割を果たした.

Malviya らは, 呼吸音を用いて COVID-19 および他の呼吸器疾患を検出するシステムを提案した [31]. 本研究では, 自己符号化器と長短期記憶ネットワーク (Long Short-Term Memory, LSTM) を組み合わせ, 音響データから疾患を高精度に分類する手法を開発している.

呼吸器疾患の診断において, 咳や呼吸音は重要なバイオマーカーとなる. 音響データを解析することで, COVID-19 や肺炎, 慢性閉塞性肺疾患 (Chronic Obstructive Pulmonary Disease, COPD) などを検出できる可能性が示されている. 特に自己符号化器は健康な音声データを学習し, 異常がある場合に高い再構築誤差を示すことで疾患の存在を検出する役割を果たす. LSTM はその後の分類フェーズで音声データの時間的特徴を学習し, 疾患の種類を判別する.

本システムは, Kaggle や Pfizer Digital Medicine Challenge から収集した呼吸音データを使用し, 515 件の音声データで構成されたデータセットで訓練・検証が実施された. データセットには, COVID-19 患者 185 名, 健康な人 150 名, およびその他の呼吸器疾患 (肺炎, COPD, 上気道感染症など) を持つ患者 180 名が含まれている. 音声データは 22,050 Hz でリサンプリングされ, 短時間フーリエ変換 (Short-Time Fourier Transform, STFT) によって特徴が抽出された.

モデルは, LSTM が 200 エポック, 自己符号化器が 100 エポックで学習され, 損失関数として平均二乗誤差が使用された. 表 2.4 の通り, 分類精度は平均 97.5% に達し, 特に COVID-19 の検出においては曲線下面積 1.0 を達成し, 感度 1.0, 特異度 0.979, F1 スコア 0.984 という非常に高い精度が得られた. 他の呼吸器疾患の分類精度も平均 96% 以上であり, COVID-19 と他の疾患の正確な識別が可能であることが示された.

表 2.4: 呼吸器疾患のマルチクラス分類の性能評価指標 ([31] より引用)

Class	Accuracy	Precision	Sensitivity	Specificity	F1-Score
Pneumonia	0.944	0.563	0.818	0.953	0.667
URTI	0.981	0.75	0.857	0.987	0.80
Bronchiolitis	0.994	1.00	0.75	1.00	0.857
<b>COVID-19</b>	<b>0.987</b>	<b>0.968</b>	<b>1.00</b>	<b>0.979</b>	<b>0.984</b>
COPD	0.944	0.981	0.867	0.990	0.920
Healthy	1.00	1.00	1.00	1.00	1.00

本研究では, 唐揚げの調理音から揚げ終わり音を検出するシステムを提案する. 唐揚げの揚げ終わり音は, 人間の耳で聞き取れる変化ではあるが, 注意を向けてないと気づくことが難しい. しかし, 唐揚げ調理音を時系列データとして捉え, 揚げ終わりの際に発生する音を異常音および変化点として捉えることが可能であるならば, 本研究においても異常音検出は有効であると考えられる.

## 2.4 画像認識による調理支援および食材の状態認識に関する研究

音だけでなく、カメラを利用した状態認識による調理支援に関する研究も実施されている。

Lin らは、人工知能（Artificial Intelligence, AI）技術を用いて食品の調理状態を自動で判定するシステムを開発した[32]。本研究では、機械ビジョンと画像処理技術を組み合わせ、食品の色の変化を解析することで「未調理」「調理済み」「過調理」の各状態を識別することを目的としている。

食品の調理状態をリアルタイムで監視するため、システムはRGB画像をHSV色空間に変換し、画像セグメンテーションを実施することで食品と背景を分離している。色相、彩度、明度の変化を利用して食品の調理成熟度や調理ムラを判断し、油の光沢やソースの吸収状態も解析する。画像処理アルゴリズムは、特定の色差や分布を用いて、調理工程を終了するか部分的に攪拌するかを判断する。

研究では、約120種類の食品データセットを用いて、調理成熟度や油の光沢、ソース吸収といったパラメータを定義し、AIアルゴリズムで解析した。実験結果では、誤認率を3%未満に抑え、各画像の解析時間を0.2秒以下に短縮することに成功した。また、キャベツ、レタス、エリンギの3種類の食品において、調理ムラや過調理箇所を特定し、自動攪拌を実行するロボットアームが実装された。

このシステムは、調理状態をより正確に判断できる食品調理の自動化を実現する可能性を示しており、将来的にはさらに多くの食品種に対応することで、食品調理の質を向上させることが期待されている。

Chao らは、バゲットの外観品質を自動評価するために、画像処理技術を活用した新しい評価システムを提案した[33]。本研究は、サイズ、平行度、色の3つの指標を基に、従来の主観的な手作業評価を置き換える客観的で定量的な評価手法を開発している。

食品業界では、食品の品質評価において外観が重要な指標となる。しかし、手作業による評価は主観的であり、評価者の熟練度に依存しやすく、時間がかかるという問題がある。本研究では、バゲットの画像を撮影し、画像処理技術を用いて自動的にバゲットの外観品質評価を行うシステムを構築した。まず、撮影画像内のベーキングトレーを基準に画像の視点を補正し、その後、バゲットの外観を評価する。

バゲット124本を対象とした実験では、サイズ評価において平均絶対誤差が0.6 mmと高精度であった。A・B・Cの三段階による平行度と色評価では、専門家の評価との一致率がそれぞれ96%および94%となり、提案手法の有効性が確認された。図2.2の左側に平行度、右側に色評価の示す。図2.2より、評価アルゴリズムと専門家の評価との間に高い整合性があることがわかる。不一致となった要素を分析すると、評価アルゴリズムの評価がより正確であり、主観的なバイアスを減らすことに役立つことがわかった。また、改良版k-meansクラスタリングを用いたベーキングトレーの抽出精度は99.21%に達し、従来手法を上回る性能を示した。

本研究は、画像処理を活用した食品品質評価の自動化の可能性を示し、手作業評価の課題を克服する新たなアプローチを提案している。今後は、深層学習技術を活用したバゲットの切り込みの評価や、より大規模なデータセットの収集により、さらなる精度向上が期待される。

Arman らは、ポテトチップスの揚げ工程をリアルタイムで最適化するスマートフライ

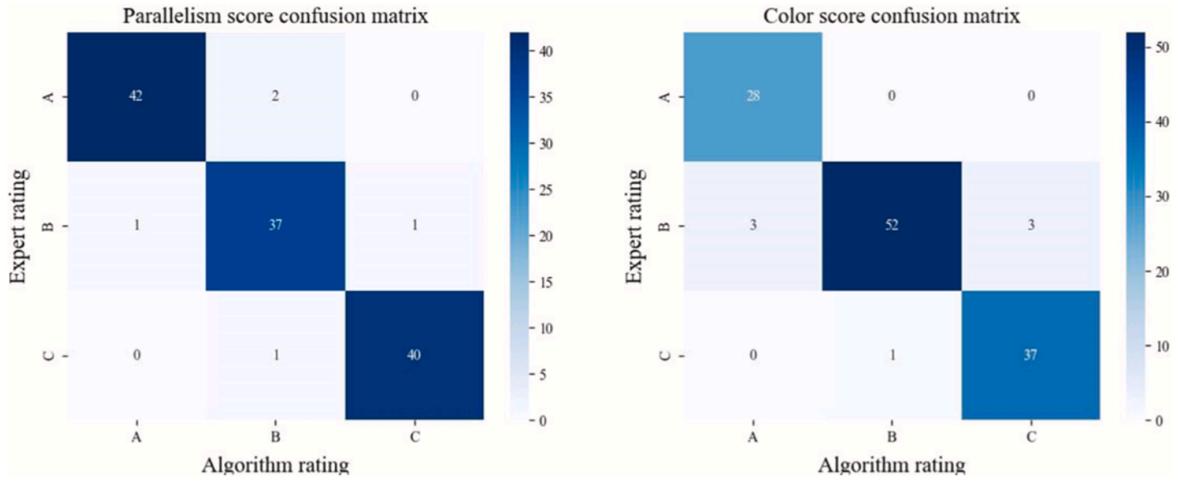


図 2.2: 平行度および色評価の混同行列 ([33] より引用)

グシステムを提案した [34]. 油吸収およびアクリルアミド生成は、健康リスクおよび品質のばらつきにつながる重要な課題である. 従来の揚げ工程では、ポテトの化学的特性の個体差が考慮されず、一律の温度・時間で調理されていた. そこで、本研究ではコンピュータビジョン、ハイパススペクトル画像、機械学習、デジタルツイン技術を統合したリアルタイム制御システムを開発することにより、リアルタイムでポテトの状態をモニタリングし、揚げ条件を動的に調整する.

ポテトを対象とした実験では、アクリルアミド生成が最大 66% 削減、油吸収量は 16% 低減された. また、色・食感の均一性が向上し、専門家の評価との一致率が高いことが確認された. リアルタイムモニタリングの精度については、色の予測精度  $R^2 = 0.96$ 、食感(硬さ)の予測精度  $R^2 = 0.91$ 、油含有量の予測精度  $R^2 = 0.98$  と高い性能を示した.

本研究は、食品加工の自動化と品質管理の高度化を実現する新たなアプローチを示している. 今後は、デジタルツイン技術を活用したさらなる最適化や、大規模データの収集により、より高度なスマートフライングシステムの開発が期待される.

このように、同じ食材において、画像分析を利用して微細な差異を検知し、分類・識別することが可能である. 本論文では、音のみを利用して、唐揚げの状態認識および調理支援を検討している. 調理環境にカメラを設置することおよび、レンズを曇らせたり、汚したりしないように注意しながら調理することは、料理初心者にとっては困難であると考えたためである. しかし、今後スマートグラスといった料理中でも、容易に利用可能なデバイスを利用して、画像と音のセンシングによるマルチモーダルシステムとして、調理支援の精度向上が実現可能であると考える.

# 第3章 FCGs: 料理初心者に向けた唐揚げ調理支援システム

本章では、料理初心者が唐揚げの不完全な調理を防ぐために、揚げ終わりを通知する料理支援システム FCGs を提案する。第1章で述べたように、音センシングは、他のセンサに比べ設置の不自由がなく簡単に利用可能であり、現代では広く利用されている技術である。よって、スマートフォンのマイクロフォンから得られる調理音から、揚げ終わりを判定する。FCGs は「Fried Chicken Goal system」の頭文字をとった造語である。

## 3.1 FCGs の提案

本節では唐揚げの揚げ終わり検出手法を提案する。唐揚げの調理音から、揚げ終わりを検出するために、機械学習および変化点検出による精度検証を実施する。その後、最も精度の良い手法を利用して、スマートフォンアプリケーションとして実装する。精度検証の結果は第5章で述べる。

### 3.1.1 唐揚げの調理状態の定義

本研究では、唐揚げの「揚げ終わり (Finish)」を衣にきつね色がしっかり付き、気泡が細かく完全に浮いている状態とし、それ以前の状態を総じて「揚げ途中 (Middle)」と定義する。

### 3.1.2 FCGs のシステム構成

本論文では、唐揚げの揚げ終わりを判別するアプリケーションの開発を目標としているため、アプリケーションのユーザビリティを考慮しないシンプルな構成とした。唐揚げのフライ調理を開始すると同時に、調理開始ボタンを押下し、揚げ終わりのフィードバックを確認した段階で調理終了ボタンを押下する。揚げ途中から揚げ終わりへの変化は、アプリケーションの画面および通知音によってフィードバックする。画面によるフィードバックは図3.1の通りである。画面上部には調理の経過時間を示すタイマーを設置している。

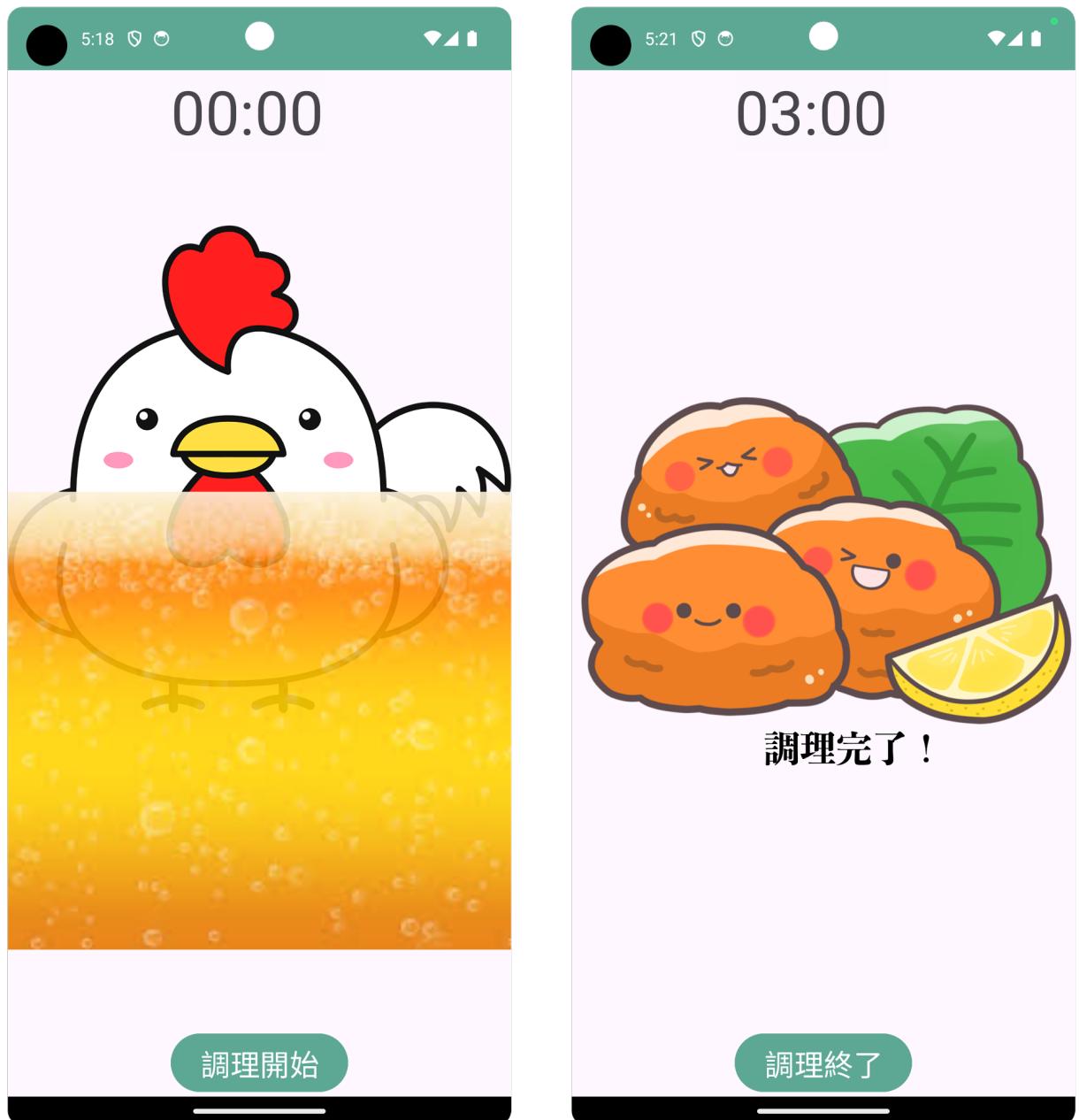


図 3.1: FCGs の画面フィードバックの様子（左図：揚げ途中，右図：揚げ終わり）

# 第4章 調理音の特徴抽出手法および識別手法

本章では本研究で使用した解析手法、特徴量、機械学習手法および変化点検出手法について述べる。また、解析はすべてプログラミング言語の Python を使用している。解析に利用したデータの詳細は表 4.1 の通りである。調理音 1 から調理音 4 は唐揚げの調理音以外の雑音（水が流れる音、キッチンタイマーの音、人の話し声等）が含まれたデータである。調理音 5 から調理音 10 は陳らが収集したデータを使用しており、静かな環境下で得られたデータである [26]。これらのデータを利用して解析および手法の検討を実施する。

表 4.1: 本研究で利用した唐揚げの調理音データ一覧

	時間 [s]	温度 [°C]	個数 [個]	油使用回数 [回]	雑音の有無
調理音 1	269	180	7	1	有
調理音 2	280	180	7	2	有
調理音 3	258	180	7	1	有
調理音 4	279	180	7	2	有
調理音 5	247	170	7	1	無
調理音 6	300	170	7	2	無
調理音 7	313	170	10	1	無
調理音 8	361	170	10	2	無
調理音 9	300	170	5	1	無
調理音 10	271	170	5	2	無

唐揚げの一般的な調理時間および調理油の温度を調査したところ、平均4分程度で170°Cから180°Cの調理油で調理していることがわかっている（表B.1）。よって、表4.1のデータも一般的な調理データであると考えられる。

## 4.1 調理音および特徴量の解析手法

本節では、調理音の解析に利用した振幅スペクトログラムおよび特徴量の解析に利用した手法を述べる。

### 4.1.1 スペクトログラム解析

本研究では、調理音データの解析に4.2.1項で後述する、振幅スペクトログラムを用いる。図4.1に示すように、振幅スペクトログラムを可視化することで、視覚的に解析することを可能にする。

この振幅スペクトログラムを用いることで、音響的な特徴の時間変化を詳細に確認でき、特定のイベントが発生する瞬間を視覚的に捉えることが可能となる。そのため、5.2.4項

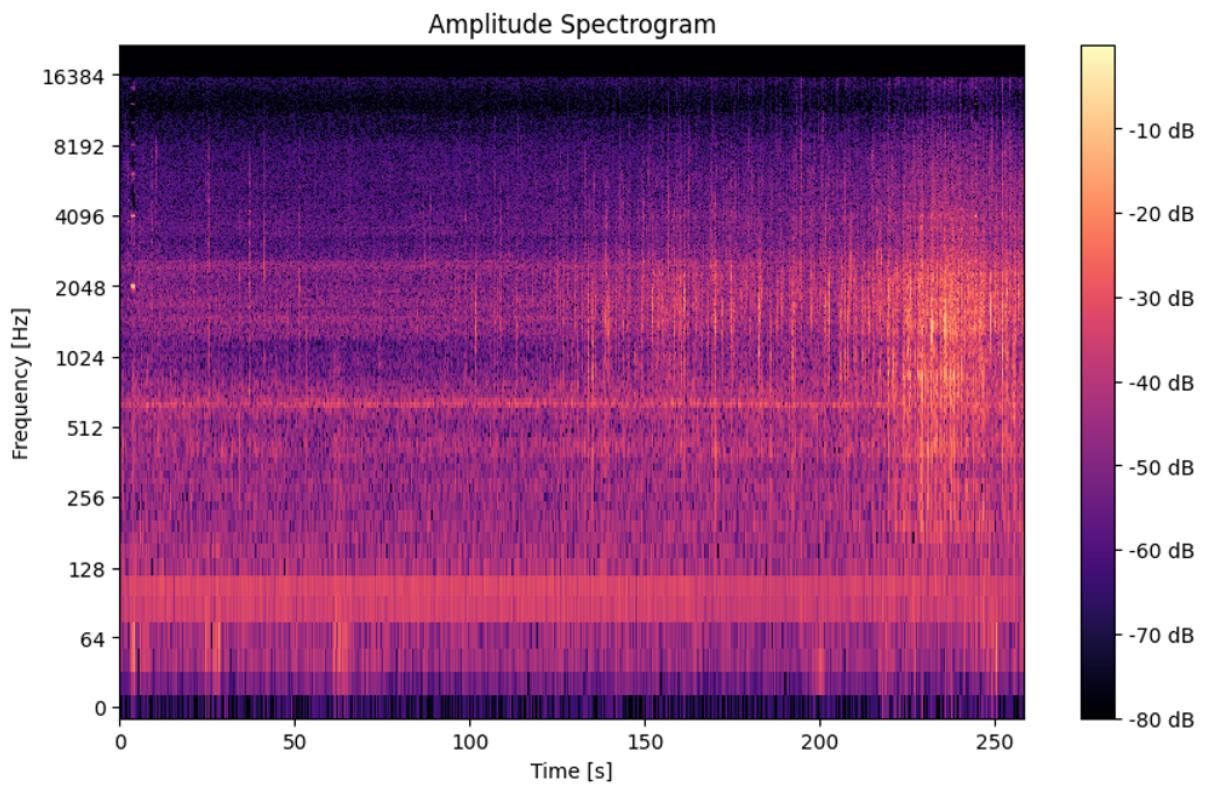


図 4.1: 調理音 3 における振幅スペクトログラムの可視化

では、振幅スペクトログラムを基に調理音の変化点を特定し、ラベリングを実施した。特に、クラスの変化が明確に反映される周波数帯域の傾向を分析し、それをラベル付けの指標として活用した。

#### 4.1.2 主成分分析

主成分分析 (Principal component analysis, PCA) は、多次元データの情報を可能な限り損なわずに低次元に圧縮する手法である [35]。PCA はデータの分散が最大となる方向を見つけ、その方向を基にデータを新しい座標軸へ変換することで次元を削減する。

高次元の特微量空間において、データの分布やクラス間の関係を直感的に理解することは困難である。PCA を用いることで、データの主要な変動方向を抽出し、低次元空間上に可視化する。

本研究では、PCA 利用して 3 次元に削減された特微量の分布を可視化する。これにより、データのクラスタリング傾向や分離性を評価する。

#### 4.1.3 t 分布型確率的近傍埋め込み法

t 分布型確率的近傍埋め込み法 (t-distributed stochastic neighbor embedding, t-SNE) は、高次元データを低次元空間（通常は 2 次元または 3 次元）に埋め込むための非線形次元削減手法である [36]。特に、データの局所的な構造を保持することに優れており、クラスタリングや特微量のパターンを視覚的に分析するために用いられる。高次元空間における

るデータ点間の近傍関係を、低次元空間上で可能な限り保持するように変換する。そのために、データ点間の類似度を確率分布として定義し、それを低次元空間に最適化する。

本研究では PCA と同様に、t-SNE を用いて特徴量の分布を可視化し、データのクラスタリング傾向や類似性を評価する。

## 4.2 調理音から抽出される音響特徴量

本節では、各種識別手法で使用する音響特徴量の算出方法を述べる。

### 4.2.1 振幅スペクトログラム

振幅スペクトログラムは、調理音データに対して STFT[37] により得られる。フーリエ変換は、時間領域の信号を周波数領域に変換し、信号の周波数成分を解析するための基本的な手法である [38]。また、任意の周期信号を複数の正弦波または余弦波に分解可能であることを示している。信号  $x(t)$  のフーリエ変換は式 4.1 で定義される。

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \quad (4.1)$$

ここで、 $X(f)$  は周波数  $f$  における信号のフーリエ変換、 $x(t)$  は時間領域での信号、 $e^{-j2\pi ft}$  はフーリエ変換の基底関数、 $t$  は時間変数、 $f$  は周波数定数である。

また、STFT は信号を短い時間区間ごとに区切ってフーリエ変換を適用する。これにより時間的な周波数変化の解析が可能となる。STFT は以下の式 4.2 で定義される。

$$X(t, f) = \int_{-\infty}^{\infty} x(\tau)h(\tau - t)e^{-j2\pi f\tau} d\tau \quad (4.2)$$

ここで、 $X(t, f)$  は時刻  $t$  における周波数  $f$  の STFT 結果、 $x(\tau)$  は時間領域の入力信号、 $h(\tau - t)$  は窓関数、 $e^{-j2\pi f\tau}$  はフーリエ変換の基底関数である。振幅スペクトログラムは式 4.2 の絶対値を取ることで求められるため、式 4.3 となる。

$$S(t, f) = |X(t, f)| \quad (4.3)$$

振幅スペクトログラムを求ることで、特定の時間帯で強く現れる周波数成分を特定可能である。図 4.2 は調理音 3 の音波形とその波形に STFT を適用した結果である。

図 4.2 の STFT を適用した結果から、0 Hz から 2,500 Hz の周波数帯の成分が大きいことがわかる。このようにして得られた振幅スペクトログラムの各時間に対して、すべての周波数の振幅の平均を計算した値を変化点検出に利用した。

### 4.2.2 MFCC

MFCC は低周波成分の解像度は高く、高周波成分は低いという人間の聴覚の周波数認識特性を反映した特徴量である [39]。音声認識および楽器の識別、調理支援など幅広い分野で利用されている。MFCC の計算手順は以下の通りである。まず、式 4.4 により窓関数を適用した音声フレーム信号  $x(n)$  に対し、DFT を適用して周波数領域のスペクトルを取得する。

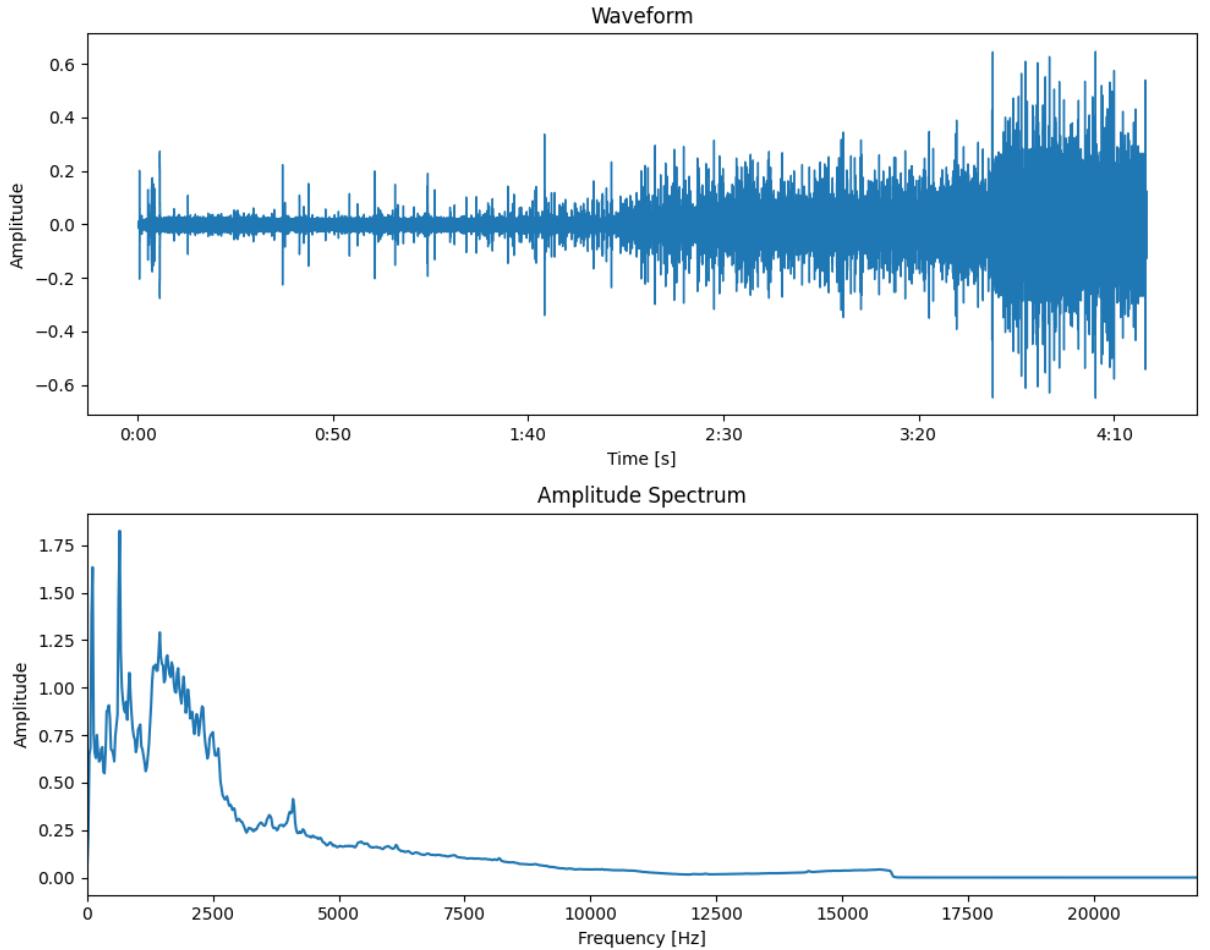


図 4.2: 調理音 3 における音波形（上）から STFT により得られた平均振幅スペクトルのプロット（下）

$$X(f) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi fn/N} \quad (4.4)$$

ここで、 $N$  はフレームの長さである。式 4.4 の二乗を取ることで、各短時間区間のパワースペクトル  $P(f)$  を求める（式 4.5）。

$$P(f) = |X(f)|^2 \quad (4.5)$$

次に人間の聴覚特性を考慮するため、メル周波数軸上で等間隔に並んだ三角窓のバンドパスフィルタで構成されるメルフィルタバンクを適用する。メル周波数とは、1,000 Hz の单一の周波数で構成された音を 1,000 mel の音高として、人間の音高知覚を考慮した尺度（メル尺度）でスケーリングした周波数である [40]。メル尺度は式 4.6 で表され、周波数  $f$  Hz をメル周波数  $M$  mel とする  $M(f)$  としてスケーリングされる。

$$M(f) = 2595 \log_{10} \left( 1 + \frac{f}{700} \right) \quad (4.6)$$

式 4.6 を適用して、メル周波数にスケーリングされたパワースペクトル  $P(M)$  をメルフィ

ルタバンクと畳み込む。臨界帯域パワースペクトルのサンプルを  $\theta(M_k)(k = 1, \dots, K)$  として式 4.7 で得る。

$$\theta(M_k) = \sum_M P(M - M_k) \psi(M) \quad (4.7)$$

ここで、 $M_k$  はメルフィルタの中心周波数、 $\psi(M)$  は臨界帯域のマスキング曲線である。計算された臨界帯域パワースペクトル  $\theta(M_k)$  に対して、式 4.8 による対数変換を適用する。

$$X(k) = \ln(\theta(M_k)), \quad k = 1, \dots, K \quad (4.8)$$

最後に離散コサイン変換 (Discrete Cosine Transform, DCT) を適用して MFCC を取得する (式 4.9)。

$$MFCC(d) = \sum_{k=1}^K X_k \cos \left[ d(k - 0.5) \frac{\pi}{K} \right], \quad d = 1, \dots, D \quad (4.9)$$

本研究では唐揚げを揚げる際に生じる調理音の変化を利用しておらず、その変化は人間の耳で十分感じ取れる音の変化である。そのため本研究では機械学習による揚げ終わり判定モデルを構築する際に、MFCC を音響特徴量として検討している。また、MFCC は、20 次元、DFT の窓サイズ 1,024 で取得する。

#### 4.2.3 ウエーブレット変換

ウェーブレット変換は、信号を異なるスケール（周波数）で分解することで、時間-周波数解析を可能にする信号処理手法である [41][42]。ウェーブレット変換は、基底関数であるマザーウエーブレットを異なるスケールと時間シフトを用いて変形しながら、信号を解析する。マザーウエーブレットは、スケールパラメータ  $a$  と時間シフトパラメータ  $b$  によって次の式 4.10 のように定義される。

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi \left( \frac{t-b}{a} \right) \quad (4.10)$$

ウェーブレット変換には、大きく分けて連続ウェーブレット変換 (Continuous Wavelet Transform: CWT) と離散ウェーブレット変換 (Discrete Wavelet Transform, DWT) の 2 種類が存在する。本研究では、計算コストが低く、信号の階層的分解が可能な DWT を用いる。

DWT は、信号を異なる周波数帯域に分解し、階層的に特徴を抽出する手法である。DWT は式 4.11 で得られる。

$$W(j, k) = \sum_{n=0}^{N-1} x(n) \psi_{j,k}(n) \quad (4.11)$$

ここで、 $x(n)$  は入力信号、 $N$  は信号の長さである。DWT の特徴の一つは、多重解像度解析によって信号を段階的に分解できることである。各分解レベルで、入力信号  $s(t)$  は式 4.12 のように表される。

$$s(t) = A_n + \sum_{i=1}^n D_i \quad (4.12)$$

ここで、 $A_n$  はレベル  $n$  の低周波成分（近似係数）、 $D_i$  は各段階の高周波成分（詳細係数）である。図 4.3 は、多重解像度解析の概念を示している。入力信号  $s$  が分解され、レベル 1 の近似係数  $A_1$  と詳細係数  $D_1$  が得られる。レベル 1 の近似係数を分解するとレベル 2 の近似係数と詳細係数が得られる。このように近似係数の分解を任意のレベルまで繰り返すことが可能である。この処理は、入力信号および近似係数に対してローパスフィルタおよびハイパスフィルタを適用することと等しい [43]。

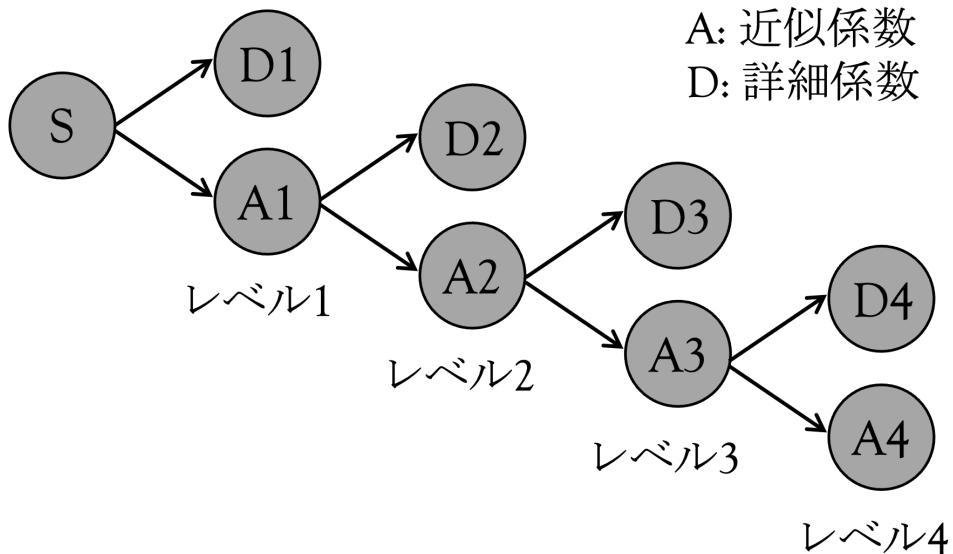


図 4.3: DWT による多重解像度解析の概念図

本研究では、機械学習による揚げ終わり判定モデルの構築の際に、レベル 10 までの近似係数および詳細係数を取得し、これらの係数から統計的特徴量を算出している。算出する統計的特徴量は表 4.2 にまとめた通りである。

これらの特徴を利用してウェーブレット変換を検討する。また、本研究では Daubechies 基底を使用してウェーブレット変換を実施する [44]。

## 4.3 調理段階の識別手法

### 4.3.1 機械学習分類手法

唐揚げの調理音から揚げ終わりを検出する手法として機械学習を検討する。本研究では教師あり学習を使用し、前節で述べた音響特徴量を検討する。機械学習は Python の機械学習ライブラリである「scikit-learn」を用いて実装する [45]。scikit-learn は、データの前処理および多種多様な機械学習手法の取り扱いが可能なライブラリである。なお、XGBoost は「xgboost」ライブラリを用いて実装する [46]。本項では、機械学習の際に検討する分類器を示す。

表 4.2: ウェーブレット変換から算出する統計的特徴量一覧

特徴量名	説明
近似係数の平均値	低周波成分の平均的なエネルギーを表す。揚げ物の音の全体的なエネルギーの変化を捉える。
詳細係数の平均値（レベル $n$ ）	レベル $n$ の詳細係数の平均。高周波成分の傾向を示し、油のはじける音の変化を測定。
標準偏差	レベル $n$ の詳細係数の散らばり具合を示す。音の変動の大きさを定量化。
最大値	レベル $n$ の詳細係数の最大値。揚げ物のピーク音を捉える。
最小値	レベル $n$ の詳細係数の最小値。音の静かな瞬間を捉える。
尖度	エネルギー分布の鋭さを示す。音の分布が尖っているかどうかを評価。
歪度	分布の左右対称性を評価し、特定の方向に偏った音の傾向を分析。
シャノンエントロピー	エネルギーの不確実性を測る指標。音のランダム性や複雑さを評価。
分散	データのばらつきを表し、音の変動の大きさを定量化する。
平均絶対値	各詳細係数の絶対値の平均を計算し、音の振幅の平均的な大きさを示す。

### サポートベクターマシン

SVM は、統計的学習理論に基づく機械学習手法の一つである [47]。異なるクラスのデータを分離する最適な超平面を求める目的とし、クラス間のマージンを最大化することによって分類する。データが線形分離できない場合には、カーネル関数を利用することで高次元空間へ写像し、線形分離可能な空間で分類する。この手法をカーネルトリックと呼ぶ [48]。本研究では、以下に示す 4 種類のカーネルを検討する。

#### 1. Linear カーネル ( $SVM_{linear}$ )

特徴空間への変換をせずにそのまま線形分類するカーネル。このカーネルを用いた場合、通常の線形分類器と類似した振る舞いをするが、マージン最大化の特性によりロバストな分類が可能となる。

#### 2. Polynominal カーネル ( $SVM_{poly}$ )

非線形の特徴空間を利用するためのカーネル。データを高次元の多項式空間にマッピングすることで、線形分離が困難なデータに対しても、非線形な分離境界を構築し、適切な分類を可能にする。

#### 3. Radial Basis Function カーネル ( $SVM_{rbf}$ )

データを無限次元の特徴空間へ非線形変換するカーネル。線形分離が困難なデータにも適用可能であり、局所的な影響を考慮し、データの分布に応じた柔軟な分類が可能である。

#### 4. Sigmoid カーネル ( $SVM_{sigmoid}$ )

ニューラルネットワークの活性化関数であるシグモイド関数に基づいたカーネル [49]. ニューラルネットワークの隠れ層に相当する特徴変換を実施する.

### ガウシアンナイーブベイズ

ガウシアンナイーブベイズ (Gaussian Naive Bayes, GNB) は単純な確率論的アルゴリズムであり、ベイズの定理を用いたナイーブベイズアルゴリズムの 1 種である [50][51]. GNB は、各クラスごとに特徴量が独立したガウス分布に従うと仮定し、事後確立を求めることで分類する。この仮定に基づき、学習データから各クラスごとの平均と分散を推定し、新しいデータに対するクラスの事後確率を計算する。

### k 近傍法

k 近傍法 (k-Nearest Neighbors, kNN) は、距離に基づく非パラメトリックな分類アルゴリズムである [52]。この手法は、未知のデータ点に対して訓練データ内の最も近い k 個のデータ点（近傍）を参照し、多数決に基づいて分類する。分類手順は以下の通りである。

1. 新しいデータ点が与えられたとき、訓練データに含まれるすべてのデータ点との距離を計算する。
2. 距離が近い上位 k 個のデータ点を選択する。
3. 選択したデータ点のクラスを多数決で決定し、最も多く含まれるクラスを新しいデータのクラスとして割り当てる。

### ランダムフォレスト

ランダムフォレストは、決定木を基にしたアンサンブル学習の一手法である [53]。この手法は複数の決定木を構築し、それらの予測を統合することで、単一の決定木に比べて過学習を抑え、頑健な分類を実現する。

### XGBoost

XGB は、勾配ブースティング (Gradient Boosting) を基盤とした機械学習手法であり、従来の勾配ブースティング決定木の改良版として開発された [54][55]。複数の決定木を逐次的に学習し、予測誤差を最小化することでモデルを最適化するブースティング手法である。各ステップで新しい決定木を作成し、既存のモデルが予測しきれなかった誤差（残差）を補正するように学習を進める。

## ロジスティック回帰

ロジスティック回帰は、線形分類モデルであり、特に二値分類問題に適している[56]。この手法は、線形回帰を拡張し、出力を確率として解釈できるようにすることで、分類タスクに適用可能としている。ロジスティック回帰では、線形回帰と同様に入力特徴量の線形結合を用いて決定境界を定めるが、出力が0または1のクラスラベルに適合するよう、シグモイド関数を用いて変換する。シグモイド関数は、入力値を0から1の範囲にマッピングする非線形関数であり、出力を確率として解釈することが可能となる。

### 4.3.2 リサンプリングによる不均等データ処理手法

本研究で扱う調理音データのクラスには大きな偏りがあったため、少数クラスに対してはオーバーサンプリング、多数クラスに対してはアンダーサンプリングを適用する。これらのリサンプリングはPythonのライブラリである「imbalanced-learn」を用いて実装する[57]。imbalanced-learnは、scikit-learnに依存しており、不均等なクラスを持つ分類を扱うためのツールを提供する。本項では、機械学習の際に検討するリサンプリング手法を示す。

#### オーバーサンプリング手法

本研究で検討する3種類のオーバーサンプリング手法を示す。

##### 1. ADASYN

ADASYN (Adaptive Synthetic Sampling) は、少数クラスに対して適応的に合成データを生成することにより、データの分布をより均衡に近づける[58]。合成データの生成は、少数クラス内の各データ点の近傍におけるデータ密度を考慮し、不均衡度が高い領域でより多くのサンプルを生成するように設計されている。

##### 2. SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) は、単純なデータのコピーではなく、少数クラスのデータ間で線形補間を実行し、新しいサンプルを生成する[59]。この手法により、データの分布をより自然に拡張し、学習モデルの汎化性能を向上させることが可能となる。本研究では、SMOTEを拡張したBorderline SMOTEおよびSVMSMOTEを検討する。

- Borderline SMOTE

Borderline SMOTE (BSMOTE) は、クラスの決定境界付近のマイノリティクラスのデータ点に重点を置いて合成データを生成することで、分類性能の向上を図る[60]。分類器の学習において、クラスの境界付近のデータは分類が特に難しく、誤分類のリスクが高いため、これらのデータを強化することで、モデルの分類性能を向上が見込まれる。

- SVMSMOTE

SVMSMOTE (Support Vector Machine Synthetic Minority Over-sampling Technique) は、SMOTEと同様に少数クラスのデータ間で合成サンプルを生成する

が、その際にSVMを用いて、分類の境界付近にあるデータ点を選択し、重点的にサンプリングする[61]。この手法により、モデルが学習する際の決定境界をより適切に調整でき、分類性能の向上が期待される。SVMSMOTEはSVM同様、4.3.1節で述べたカーネルを選択できるため、同様に4種類のカーネルを検討する。

### アンダーサンプリング手法

本研究で検討する5種類のアンダーサンプリング手法を示す。

#### 1. Cluster Centroids

ClusterCentroids(CC)は、多数クラスのデータをクラスタリングし、各クラスタの重心を代表点としてサンプリングすることで、データのバランスを調整する手法である[62]。この手法は、データの情報をなるべく保持しつつ、多数クラスのデータサイズを削減することを目的としている。

#### 2. Condensed Nearest Neighbour

Condensed Nearest Neighbour(CNN)は、分類に必要なデータ点を保持しつつ、冗長なデータ点を削除する[63]。少数クラスのデータをそのまま保持し、多数クラスのデータからランダムに1つのデータ点を選択し、初期プロトタイプセット $S$ を作成する。残りの多数クラスのデータを、 $S$ を用いて1-NN分類を実施する。もし誤分類された多数クラスのデータ点が存在すれば、それを $S$ に追加する。すべての多数クラスのデータ点が正しく分類されるまで、このプロセスを繰り返すことにより、分類の決定境界を形成するために必要なデータ点のみを保持し、不必要的データ点の削除が可能となる。

#### 3. Near Miss

NearMissは、多数クラスのデータ点を削減する際に、少数クラスとの近傍関係を考慮することで、適切なデータ削減を実現する[64]。NearMisssにはバージョン1から3の3種類存在し、それぞれ異なる基準で多数クラスのデータを削減する。

- NearMiss-1 (NM1)

各多数クラスのデータ点について、少数クラスのデータ点までの距離を計算する。次に、少数クラスの $k$ 個の最近傍データ点との距離の平均を求める。距離の平均が最も小さい多数クラスのデータ点を保持し、それ以外のデータを削除する。

- NearMiss-2 (NM2)

各多数クラスのデータ点について、少数クラスのデータ点までの距離を計算する。次に、少数クラスの $k$ 個の最近傍データ点との距離の平均を求める。距離の平均が最も大きい多数クラスのデータ点を保持し、それ以外のデータを削除する。

- NearMiss-3 (NM3)

各少数クラスのデータ点に対して、多数クラスのデータ点との距離を計算する。

最も近いN個の多数クラスのデータ点のみを保持し、それ以外の多数クラスのデータを削除する。

本研究では、3種類のバージョンのNearMissを検討する。

#### 4. Neighbourhood Cleaning Rule

Neighbourhood Cleaning Rule (NCR) は、データセット内のノイズや誤分類の可能性が高いデータを削除することで、分類モデルの精度を向上させる手法である[65]。NCRの手順は、まず、各データに対してkNNを適用し、近傍データ点のクラス分布を分析する。特定のデータ点が、近傍の多数派クラスと異なるクラスである場合、そのデータ点は「ノイズ」または「誤分類される可能性が高いデータ」とみなされる。次に、近傍の多数クラスのデータ点に囲まれた少数クラスのデータは保持し、近傍のデータ点と一致しない多数クラスのデータ点を削除する。最後に、決定境界を維持するため、近傍の多数クラスのデータと一致しない少数クラスのデータを削除する。

#### 5. One-Sided Selection

One-Sided Selection (OSS) は、多数クラスの冗長なデータを削減しつつ、少数クラスの重要なデータを保持する手法である[66]。これは、CNNとTomek Linksという2つのアプローチを組み合わせることで実現されている。Tomek Linksは、異なるクラスに属する2つのデータ点が互いに最近傍である場合、それらのデータ点がクラスの境界付近に位置することを示す[67]。OSSでは、Tomek Linksのペアのうち、多数クラスのデータ点を削除する。これにより、決定境界の近くにある多数クラスのデータ点を削減する。その後、CNNを適用し、多数クラスのデータセットを縮小する。このプロセスにより、ノイズや冗長なデータを削減しながら、分類の決定境界を明確化することが可能となる。

本研究では、本項で述べたオーバーサンプリング手法およびアンダーサンプリング手法を組み合わせて、不均等データ処理を実施する。

#### 4.3.3 調理音の変化点検出手法

変化点検出は、時系列データにおいて統計的な性質が変化する点を特定する技術である。本研究では、唐揚げの揚げ終わりの音を変化点として捉え、調理過程における音響特徴の変化を解析することで、適切な揚げ時間の検出が可能であるかを検討する。変化点検出はPythonのライブラリである「ruptures」を用いて実装する[68]。rupturesは、非定常信号の分析とセグメンテーションのためのメソッドが取り扱い可能なライブラリである。

本研究では、変化点検出手法の一つであるWindowアルゴリズムを利用する[69]。この手法は、データの局所的な変化を捉えやすく、リアルタイムの変化点検出に適している。Windowアルゴリズムは、データストリームに沿ってスライドする2つのウィンドウを用いて、それぞれのウィンドウ内の信号の統計的特性を比較し、差異を計測することで変化点を検出する。与えられたコスト関数 $c$ に対して、区間 $a, b$ における信号 $y$ の差異を測る指標 $d$ は式4.13で定義される。

$$d(y_{a.t}, y_{t.b}) = c(y_{a.b}) - c(y_{a.t}) - c(y_{t.b}), \quad (a < t < b) \quad (4.13)$$

本研究では、マハラノビス距離をコスト関数として利用する。マハラノビス距離は、データの共分散構造を考慮した距離計量を計算する。マハラノビス型の距離を使用したコスト関数  $c_M$  は式 4.14 のように定義される。

$$c_M(y_{a.b}) = \sum_{t=a+1}^b \|y_t - \bar{y}_{a.b}\| M^2 \quad (4.14)$$

ここで、 $\bar{y}_{a,b}$  は部分信号  $y_{a,b}$  の平均値である。マハラノビス距離は、通常のユークリッド距離よりも特徴を適切に捉えることが可能である。

変化点検出では、コスト関数を最小化することで変化点を決定する。その際、式 4.15 で与えられるペナルティ項を加えた目的関数を最小化する。

$$TotalCost = \sum_{i=0}^k c_M(y_{t_i, t_{i+1}}) + \lambda \cdot k \quad (4.15)$$

ここで、 $c_M(y_{t_i, t_{i+1}})$  は各セグメント  $t_i, t_{i+1}$  間におけるコスト関数、 $k$  は検出された変化点の数、 $\lambda$  はペナルティの重みである。ペナルティが大きいほど、変化点の数が減少し、小さいほど変化点が増加する。よって、適切なペナルティを設定する必要がある。

本研究では、前述した機械学習手法に加え、本変化点検出手法の精度を比較し、最も精度の良い手法をアプリケーションに実装する。

#### 4.3.4 調理音に含まれる雑音の処理

調理音には、お皿がぶつかる音やタイマーの音など突発的な雑音が含まれており、目立っている。そこで本研究では、メディアンフィルタを使用する[70]。メディアンフィルタは、突発的な大きなノイズの除去に有効である。メディアンフィルタの例を図4.4に示す。データの集合を降順に並べ、中心の位置の値（中央値）を出力する。これにより、突発的な値を除去しつつ、トレンドの変化を合理的に維持することができる。

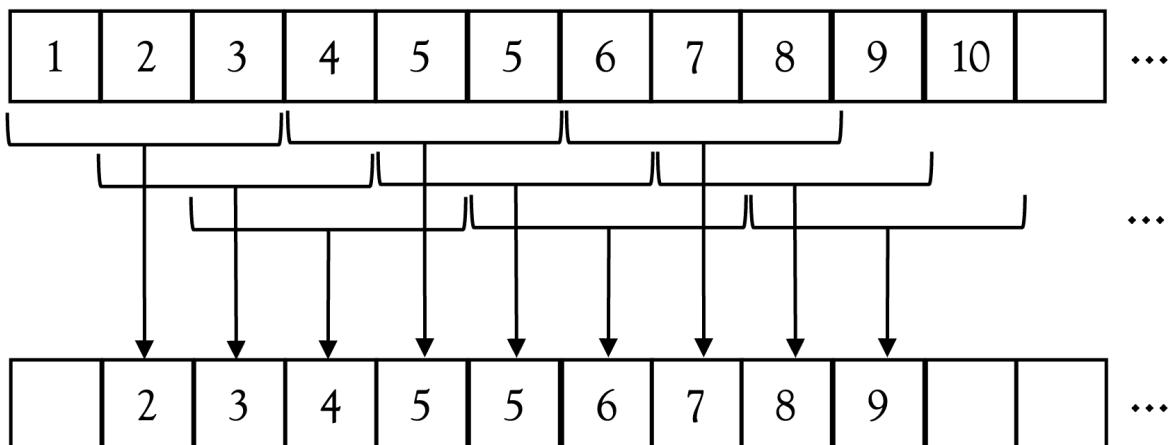


図 4.4: メディアンフィルタの仕組み

メディアンフィルタは、変化点検出を実施する際に、振幅の平均に対して適用する。

## 4.4 各識別手法の評価方法および評価指標

本節では、本研究で使用する評価方法および評価指標の4種類について述べる。

### 4.4.1 一グループ抜き交差検証による評価方法

まず評価方法について、本研究では唐揚げのフライ調理音10種類に対して、一グループ抜き交差検証(Leave-One-Group-Out Cross-Validation, LOGO-CV)を実施する[71]。LOGO-CVは、データを予め定義したグループ単位で分割し、各グループを順番にテストデータとして取り除く手法である。本研究におけるLOGO-CVを図4.5に示す。本研究の場合は、データは調理音ごとに分けられているため、各調理音を一つのグループと見なし、グループごとに交差検証を実施する。

The diagram illustrates the Leave-One-Group-Out Cross-Validation (LOGO-CV) process for 10 cooking sounds (調理音1 to 調理音10) across 10 trials (1回目 to 10回目). A legend indicates that white boxes represent '学習データ' (Learning Data) and blue boxes represent 'テストデータ' (Test Data). In each trial, one specific cooking sound is highlighted in blue, indicating it is the test data for that iteration, while all other sounds are shown in white, indicating they are learning data.

回目	調理音1	調理音2	調理音3	調理音4	調理音5	調理音6	調理音7	調理音8	調理音9	調理音10
1回目	調理音1	調理音2	調理音3	調理音4	調理音5	調理音6	調理音7	調理音8	調理音9	調理音10
2回目	調理音1	調理音2	調理音3	調理音4	調理音5	調理音6	調理音7	調理音8	調理音9	調理音10
3回目	調理音1	調理音2	調理音3	調理音4	調理音5	調理音6	調理音7	調理音8	調理音9	調理音10
4回目	調理音1	調理音2	調理音3	調理音4	調理音5	調理音6	調理音7	調理音8	調理音9	調理音10
5回目	調理音1	調理音2	調理音3	調理音4	調理音5	調理音6	調理音7	調理音8	調理音9	調理音10
6回目	調理音1	調理音2	調理音3	調理音4	調理音5	調理音6	調理音7	調理音8	調理音9	調理音10
7回目	調理音1	調理音2	調理音3	調理音4	調理音5	調理音6	調理音7	調理音8	調理音9	調理音10
8回目	調理音1	調理音2	調理音3	調理音4	調理音5	調理音6	調理音7	調理音8	調理音9	調理音10
9回目	調理音1	調理音2	調理音3	調理音4	調理音5	調理音6	調理音7	調理音8	調理音9	調理音10
10回目	調理音1	調理音2	調理音3	調理音4	調理音5	調理音6	調理音7	調理音8	調理音9	調理音10

図 4.5: 本研究における一グループ抜き交差検証

### 4.4.2 評価指標

性能評価には、適合率、再現率、F1スコア、および正解率を使用する。機械学習モデルの性能評価においては、揚げ途中と揚げ終わりの2クラス分類におけるモデルの予測結果を、実際のクラスと比較したものを混同行列として表す。混同行列の構造を表4.3に示す。

表 4.3: 混同行列の構造

	実際のクラス 0 (Negative)	実際のクラス 1 (Positive)
予測クラス 0 (Negative)	TN (True Negative)	FN (False Negative)
予測クラス 1 (Positive)	FP (False Positive)	TP (True Positive)

本研究においては、クラス0が揚げ途中、クラス1が揚げ終わりを意味する。各要素の意味は以下の通りである。

### TN (True Negative, 真陰性)

実際のクラスが揚げ途中 (Negative) であり, モデルが正しく揚げ途中と予測した数.

### TP (True Positive, 真陽性)

実際のクラスが揚げ終わり (Positive) であり, モデルが正しく揚げ終わりと予測した数.

### FP (False Positive, 偽陽性)

実際のクラスが揚げ途中 (Negative) であるにもかかわらず, モデルが誤って揚げ終わりと予測した数.

### FN (False Negative, 偽陰性)

実際のクラスが揚げ終わり (Positive) であるにもかかわらず, モデルが誤って揚げ途中と予測した数.

次に, 各評価指標について述べる. 適合率は, モデルが揚げ終わり (Positive) と予測したデータのうち, 実際に揚げ終わりであったデータの割合を示す. 数式は以下の通りである :

$$Precision = \frac{TP}{TP + FP} \quad (4.16)$$

適合率が高いほど, 揚げ終わりと予測したデータの中に誤って揚げ途中が分類される割合が少ないことを意味する. 本研究では, 安全な揚げ終わり判定に重点を置いているため, 適合率を重視する.

再現率は, 実際に揚げ終わり (Positive) であるデータのうち, モデルが正しく揚げ終わりと予測した割合を示す. 数式は以下の通りである :

$$Recall = \frac{TP}{TP + FN} \quad (4.17)$$

再現率が高いほど, 実際に揚げ終わりであるデータを取りこぼさずに予測できていることを意味する.

F1スコアは, 適合率と再現率の調和平均であり, バランスの取れた評価指標である. 数式は以下の通りである :

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4.18)$$

F1スコアは, 適合率と再現率のバランスが重要な場合に用いられる.

正解率は, 全予測のうち正しく分類された割合を示す. 数式は以下の通りである :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.19)$$

正解率は全体的なモデルの性能を示すが, クラスの分布が不均衡な場合には注意が必要である.

# 第5章 調理音識別手法およびFCGsの開発

本章では第4章で述べた解析手法を利用した唐揚げの揚げ終わり分類および検出の精度評価について述べる。また、最も精度の良い手法を利用して、リアルタイムで揚げ終わりを識別するスマートフォンアプリケーションを開発する。

## 5.1 調理音データのラベリング

調理音データのラベリングは、音声解析ソフト ELAN を使用した[72]。本研究では、3.1.1 項で述べた通り、ラベルの種類を「揚げ途中」と「揚げ終わり」の2クラスとした。ラベルの基準は実際の音、衣の色の変化、油の気泡の変化、ELAN による波形解析を利用して決定した。

各調理音に対するクラスごとのデータ時間は 5.1 のようになっている。

表 5.1: ラベル付けしたデータのラベル時間

	調理時間 [s]	揚げ途中 [s]	揚げ終わり [s]
調理音 1	269	243	26
調理音 2	280	240	40
調理音 3	258	224	34
調理音 4	279	253	26
調理音 5	247	236	11
調理音 6	300	290	10
調理音 7	313	302	10
調理音 8	361	340	21
調理音 9	300	285	15
調理音 10	271	259	12

揚げ途中から揚げ終わりへの変化の様子は図 5.1 のようになっている。揚げ終わりに近くにつれて衣の色がきつね色になり、気泡も細かくなっていることがわかる。

## 5.2 機械学習モデルによる唐揚げの調理音データ分類精度の評価

本節では、第4章で述べた手法を利用して、唐揚げの調理音から揚げ途中および揚げ終わりの分類が可能な機械学習モデルを検討する。表 5.1 の通り、揚げ途中と揚げ終わりクラスのデータバランスは不均等である。よって、オーバーサンプリングおよびアンダーサンプリングの両手法を適用し、データセットを均等にする。また、分類モデルは 1 調理音抜き交差検証を実施し、F1 スコア、適合率、再現率、正解率の 4 種類の指標で評価する。使用する調理音は 10 種類であるため、各指標において、10 回分の平均値および標準偏差を示す。



図 5.1: 揚げ途中から揚げ終わりへの変化の様子

標準偏差を算出して、分類精度として評価する。分類は9種類の分類器を使用して実施し、アンダーサンプリング手法7種、オーバーサンプリング手法6種、分類器9種の378通りをMFCC、ウェーブレット変換から得られる特徴をそれぞれ音響特徴量として利用し検証する。

機械学習の際、窓幅は1秒、シフト幅は0.25秒としており、各調理音のサンプリング周波数は44.1 kHzである。真値ラベルは、フレームの半分以上が揚げ終わりのラベル時間を含む場合は「揚げ終わり」、そうでない場合は「揚げ途中」とラベル付けされた。

### 5.2.1 MFCC 特徴量を使用した分類精度の評価

本項では、調理音からMFCCを特徴量として抽出した場合の分類精度を評価する。まず、両クラスの割合を5:5にした場合における、最も高い精度を含むNCRとの組み合わせの分類精度を以下に示す。リサンプリングの順序は、アンダーサンプリングを適用した後、オーバーサンプリングを適用している。表5.2はオーバーサンプリングにBSMOTEを適用した各分類器による分類精度である。リサンプリング適用前のクラスのサンプル数の比は Middle: Finish = 10,608: 870 であり、適用後の比は、Middle: Finish = 10,449: 10,449 となった。

表 5.2: NCR+BSMOTE を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.521 \pm 0.345$	$0.551 \pm 0.312$	$0.591 \pm 0.424$	$0.622 \pm 0.150$
ランダムフォレスト	$0.105 \pm 0.196$	$0.847 \pm 0.312$	$0.071 \pm 0.141$	$0.527 \pm 0.062$
GNB	$0.488 \pm 0.424$	$0.606 \pm 0.356$	$0.597 \pm 0.512$	$0.646 \pm 0.179$
kNN	$0.207 \pm 0.308$	$0.456 \pm 0.435$	$0.207 \pm 0.353$	$0.523 \pm 0.123$
$SVM_{linear}$	$0.510 \pm 0.351$	$0.524 \pm 0.319$	$0.584 \pm 0.433$	$0.614 \pm 0.157$
$SVM_{poly}$	$0.520 \pm 0.379$	$0.690 \pm 0.293$	$0.619 \pm 0.476$	$0.634 \pm 0.172$
$SVM_{rbf}$	$0.539 \pm 0.355$	$0.625 \pm 0.268$	$0.646 \pm 0.461$	$0.640 \pm 0.151$
$SVM_{sigmoid}$	$0.468 \pm 0.410$	$0.619 \pm 0.373$	$0.563 \pm 0.497$	$0.587 \pm 0.224$
XGB	$0.178 \pm 0.267$	$0.568 \pm 0.431$	$0.151 \pm 0.242$	$0.539 \pm 0.065$

F1スコアと適合率がともに6割を超えているモデルは見受けられなかった。また、5割を超えているモデルにおいても、標準偏差に注目すると安定性が低いことわかる。表5.3

はオーバーサンプリングに  $SVMSMOTE_{linear}$  を適用した各分類器による分類精度である。リサンプリング後のサンプル比は  $Middle: Finish = 10,449:8,898$  となった。

表 5.3: NCR+ $SVMSMOTE_{linear}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.506 \pm 0.294$	$0.596 \pm 0.264$	$0.588 \pm 0.419$	$0.636 \pm 0.129$
ランダムフォレスト	$0.059 \pm 0.122$	$0.701 \pm 0.382$	$0.037 \pm 0.081$	$0.557 \pm 0.074$
GNB	$0.394 \pm 0.411$	$0.623 \pm 0.358$	$0.506 \pm 0.520$	$0.614 \pm 0.181$
kNN	$0.157 \pm 0.253$	$0.411 \pm 0.398$	$0.158 \pm 0.314$	$0.536 \pm 0.098$
$SVM_{linear}$	$0.549 \pm 0.265$	$0.591 \pm 0.250$	$0.643 \pm 0.386$	$0.645 \pm 0.107$
$SVM_{poly}$	<b><math>0.576 \pm 0.324</math></b>	<b><math>0.729 \pm 0.193</math></b>	<b><math>0.701 \pm 0.417</math></b>	<b><math>0.678 \pm 0.130</math></b>
$SVM_{rbf}$	$0.546 \pm 0.352$	$0.617 \pm 0.270$	$0.699 \pm 0.463$	$0.668 \pm 0.138$
$SVM_{sigmoid}$	$0.383 \pm 0.381$	$0.541 \pm 0.371$	$0.515 \pm 0.512$	$0.542 \pm 0.225$
XGB	$0.071 \pm 0.110$	$0.446 \pm 0.421$	$0.058 \pm 0.106$	$0.527 \pm 0.081$

BSMOTE と同様に、F1 スコアと適合率がともに 6 割を超えているモデルはなかった。しかし、 $SVM_{poly}$  は F1 スコアが  $57.6\% \pm 32.4\%$  ではあるが、適合率が 7 割を超えて安定していることがわかる。表 5.4 はオーバーサンプリングに  $SVMSMOTE_{poly}$  を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle: Finish = 10,449:10,449$  である。

表 5.4: NCR+ $SVMSMOTE_{poly}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.574 \pm 0.304$	$0.614 \pm 0.259$	$0.639 \pm 0.396$	$0.649 \pm 0.135$
ランダムフォレスト	$0.138 \pm 0.227$	$0.729 \pm 0.401$	$0.100 \pm 0.166$	$0.531 \pm 0.064$
GNB	$0.486 \pm 0.405$	$0.649 \pm 0.302$	$0.591 \pm 0.498$	$0.634 \pm 0.170$
kNN	$0.218 \pm 0.317$	$0.460 \pm 0.435$	$0.226 \pm 0.370$	$0.516 \pm 0.136$
$SVM_{linear}$	$0.559 \pm 0.311$	$0.588 \pm 0.264$	$0.626 \pm 0.398$	$0.634 \pm 0.142$
$SVM_{poly}$	$0.535 \pm 0.363$	$0.705 \pm 0.261$	$0.629 \pm 0.459$	$0.637 \pm 0.166$
$SVM_{rbf}$	$0.574 \pm 0.341$	$0.645 \pm 0.270$	$0.691 \pm 0.441$	$0.656 \pm 0.136$
$SVM_{sigmoid}$	$0.455 \pm 0.401$	$0.614 \pm 0.369$	$0.549 \pm 0.497$	$0.576 \pm 0.219$
XGB	$0.252 \pm 0.297$	$0.462 \pm 0.414$	$0.217 \pm 0.277$	$0.560 \pm 0.086$

NCR+ $SVMSMOTE_{poly}$  を適用すると、NCR+ $SVMSMOTE_{linear}$  同様、 $SVM_{poly}$  分類器の F1 スコアが 5 割以上かつ適合率 7 割以上のモデルであることがわかる。表 5.5 はオーバーサンプリングに  $SVMSMOTE_{rbf}$  を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle: Finish = 10,449:10,449$  である。

結果、rbf カーネルを使用した SVMSMOTE は、poly カーネルを使用している SVMSMOTE と同様の精度が得られたことがわかる。表 5.6 はオーバーサンプリングに  $SVMSMOTE_{sigmoid}$  を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle: Finish = 10,449:10,449$  である。

表 5.6 の通り、sigmoid カーネルを使用した SVMSMOTE も、他のカーネルを使用した SVMSMOTE と同様の結果となった。表 5.7 はオーバーサンプリングに ADASYN を適用し

表 5.5: NCR+ $SVM_{SMOTE_{rbf}}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	0.570 ± 0.298	0.613 ± 0.260	0.632 ± 0.391	0.644 ± 0.130
ランダムフォレスト	0.146 ± 0.246	0.725 ± 0.401	0.112 ± 0.196	0.536 ± 0.072
GNB	0.487 ± 0.403	0.653 ± 0.298	0.591 ± 0.497	0.633 ± 0.171
kNN	0.219 ± 0.320	0.457 ± 0.433	0.228 ± 0.375	0.516 ± 0.140
$SVM_{linear}$	0.558 ± 0.309	0.586 ± 0.265	0.627 ± 0.396	0.631 ± 0.141
$SVM_{poly}$	0.528 ± 0.363	0.702 ± 0.264	0.621 ± 0.460	0.633 ± 0.164
$SVM_{rbf}$	0.568 ± 0.343	0.642 ± 0.268	0.684 ± 0.442	0.652 ± 0.134
$SVM_{sigmoid}$	0.456 ± 0.400	0.614 ± 0.370	0.549 ± 0.496	0.576 ± 0.219
XGB	0.217 ± 0.267	0.457 ± 0.382	0.177 ± 0.234	0.544 ± 0.081

表 5.6: NCR+ $SVM_{SMOTE_{sigmoid}}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	0.561 ± 0.299	0.609 ± 0.261	0.622 ± 0.390	0.638 ± 0.128
ランダムフォレスト	0.144 ± 0.240	0.681 ± 0.390	0.110 ± 0.195	0.532 ± 0.071
GNB	0.491 ± 0.399	0.665 ± 0.293	0.593 ± 0.495	0.634 ± 0.171
kNN	0.219 ± 0.321	0.457 ± 0.433	0.228 ± 0.375	0.516 ± 0.140
$SVM_{linear}$	0.550 ± 0.309	0.583 ± 0.266	0.616 ± 0.396	0.626 ± 0.140
$SVM_{poly}$	0.527 ± 0.364	0.701 ± 0.266	0.620 ± 0.461	0.632 ± 0.165
$SVM_{rbf}$	0.568 ± 0.343	0.642 ± 0.268	0.685 ± 0.442	0.653 ± 0.135
$SVM_{sigmoid}$	0.456 ± 0.400	0.614 ± 0.370	0.549 ± 0.496	0.577 ± 0.219
XGB	0.217 ± 0.287	0.486 ± 0.440	0.186 ± 0.256	0.546 ± 0.075

た各分類器による分類精度である。クラスのサンプル数は  $Middle: Finish = 10,449: 10,428$  である。

結果として、NCR に対しては、どのオーバーサンプリング手法を組み合わせても、同じような結果が得られた。その中でも僅差で NCR+ $SVM_{SMOTE_{linear}}$  の組み合わせにおける、 $SVM_{poly}$  分類器を使用した際の分類精度が、最も良い結果となった。しかし、全体として F1 スコアと適合率が 8 割以上の分類精度となるモデルは構築できなかった。

原因として、以下の 3 点が考えられる。

1 つは、オーバーサンプリング、特に SMOTE や ADASYN は、少数クラスのデータを合成するが、既存のデータポイントを補完しているだけであり、新しい情報を追加するわけではない。そのため、訓練データに過度に適合し、汎化性能が低下した可能性がある。

2 つ目として、NCR+ $SVM_{SMOTE_{linear}}$  の組み合わせにおいて、元のデータセットのクラス比である 10,608:870 から 10,449:8,898 にリサンプリングしたように、本来のデータ分布と大きく異なるデータで学習した。これにより、クラスの決定境界が不安定になった可能性が考えられる。

3 つ目として、F1 スコアのばらつきが大きいことからも、特徴量が十分にクラスを分離できていない可能性が考えられる。

よって、PCA および t-SNE による特徴の可視化を実施し、クラス間の分離が適切であ

表 5.7: NCR+ADASYN を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.553 \pm 0.320$	$0.609 \pm 0.270$	$0.618 \pm 0.418$	$0.645 \pm 0.139$
ランダムフォレスト	$0.119 \pm 0.217$	$0.839 \pm 0.311$	$0.085 \pm 0.165$	$0.533 \pm 0.066$
GNB	$0.475 \pm 0.415$	$0.614 \pm 0.367$	$0.572 \pm 0.498$	$0.634 \pm 0.174$
kNN	$0.193 \pm 0.309$	$0.418 \pm 0.405$	$0.203 \pm 0.363$	$0.509 \pm 0.131$
$SVM_{linear}$	$0.532 \pm 0.303$	$0.572 \pm 0.271$	$0.594 \pm 0.402$	$0.615 \pm 0.136$
$SVM_{poly}$	$0.531 \pm 0.377$	$0.691 \pm 0.295$	$0.632 \pm 0.469$	$0.639 \pm 0.169$
$SVM_{rbf}$	$0.570 \pm 0.341$	$0.643 \pm 0.269$	$0.684 \pm 0.441$	$0.655 \pm 0.137$
$SVM_{sigmoid}$	$0.446 \pm 0.399$	$0.611 \pm 0.368$	$0.538 \pm 0.500$	$0.573 \pm 0.216$
XGB	$0.083 \pm 0.129$	$0.604 \pm 0.434$	$0.053 \pm 0.082$	$0.493 \pm 0.059$

るかを確認する。

### 5.2.2 PCA および t-SNE による MFCC 特徴量解析

前項で最も分類精度の良い分類器を含んでいた、NCR+ $SVM_{SMOTE_{linear}}$  のリサンプリング手法前後を解析し、リサンプリングの影響および MFCC のクラス分離能力を確認する。

図 5.2 は PCA の結果である。左図がリサンプリング適用前で、右図はリサンプリング適用後である。青色が揚げ途中で、赤色は揚げ終わりを示している。

リサンプリング前は、揚げ途中が圧倒的に多く、揚げ途中は少数点在していることがわかる。また、揚げ途中は 3 つほどの異なるクラスタに分かれており、データ構造に一定のパターンがあるように見える。揚げ終わりは、揚げ途中のデータの間に少しづつ散らばっており、明確なクラスタを形成していないことがわかる。このことから、揚げ終わりは明確な特徴を持たない、もしくは揚げ途中の中に埋もれており、単純な決定境界による分類が困難であると考えられる。また、揚げ途中がいくつかのクラスタを形成していることから、揚げ途中内部に異なるサブクラスがあると推測される。

これに対して、リサンプリング後は、揚げ終わりのデータは増加したが、データの分布は大きく変化しておらず、揚げ途中のデータ構造もリサンプリング前と大きな変化が見られないことがわかる。このことから、揚げ途中と揚げ終わりの分離が改善されておらず、揚げ終わりのデータ数が単に増加しただけとなっている可能性が考えられる。

次に、t-SNE による解析結果を図 5.3 に示す。左図がリサンプリング適用前で、右図はリサンプリング適用後である。

リサンプリング前は、揚げ途中が圧倒的に多く、揚げ終わりはデータの端や一部にまとまって分布していることがわかる。また、揚げ終わりは明確なクラスタを形成せず、揚げ途中のデータの間に散らばっている。

これに対して、リサンプリング後は揚げ終わりのデータが大幅に増加し、データの形は曲線状になっており、不自然な分布が見られる。

原因として、揚げ終わりのサンプルを人工的に増やしたことにより、歪んだデータ構造になった可能性が高い。この曲線状になった分布が、モデルの学習に悪影響を及ぼしてい

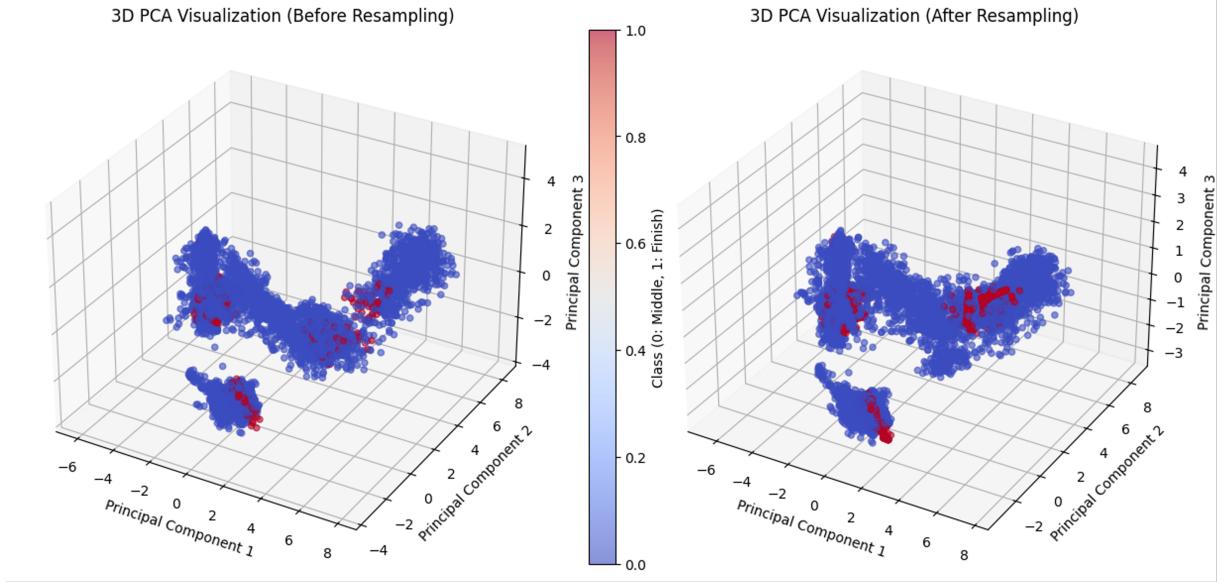


図 5.2: PCA による MFCC 解析結果 (NCR+*SVM**SMOTE*<sub>linear</sub> 適用前後)

る可能性が考えられる。特に、オーバーサンプリングにより、データバランスを 5:5 にしたため、決定境界付近でデータが過剰合成されていると考えられる。

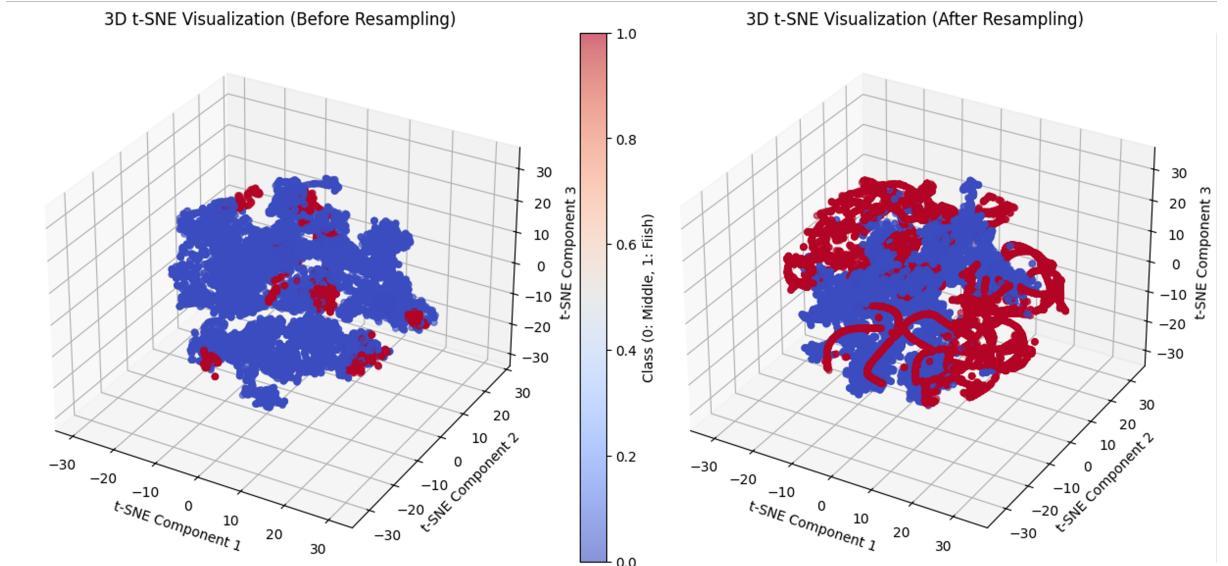


図 5.3: t-SNE による MFCC 解析結果 (NCR+*SVM**SMOTE*<sub>linear</sub> 適用前後)

以上の結果より、クラスのデータバランスを調整する必要があると考える。そのため、リサンプリングのバランスを 5:5 ではなく、6:4 のように調整して、再度モデルの性能を評価する。また、揚げ途中クラスの細分化を実施し、データバランスの不均等を緩和する。

### 5.2.3 リサンプリングバランスの調整による分類モデルの再評価

本項では、データバランスを 6:4 となるようにリサンプリングを適用し、機械学習モデルを検証する。ここで、アンダーサンプリング手法の内、リサンプリングの割合を指定で

きないアルゴリズムである CNN, NCR, OSS は検討対象外とした.

#### 5.2.4 振幅スペクトログラムに基づく再ラベリング

揚げ途中クラスを細分化し、データセットの不均等を緩和するために、再度ラベリングを実施する。ラベリングは主に、振幅の強度に基づいて実施し、振幅の強度は調理音の振幅スペクトログラムを可視化することで視覚的に捉える。ラベリング後は、実際の調理音、調理時に撮影されていた動画を確認して、各ラベルの整合性が取れているか確認する。

ラベリングの結果、各調理音は4クラスに分けられた。図5.4は、調理音3におけるラベリング結果である。白線はクラスが切り替わる瞬間を示している。

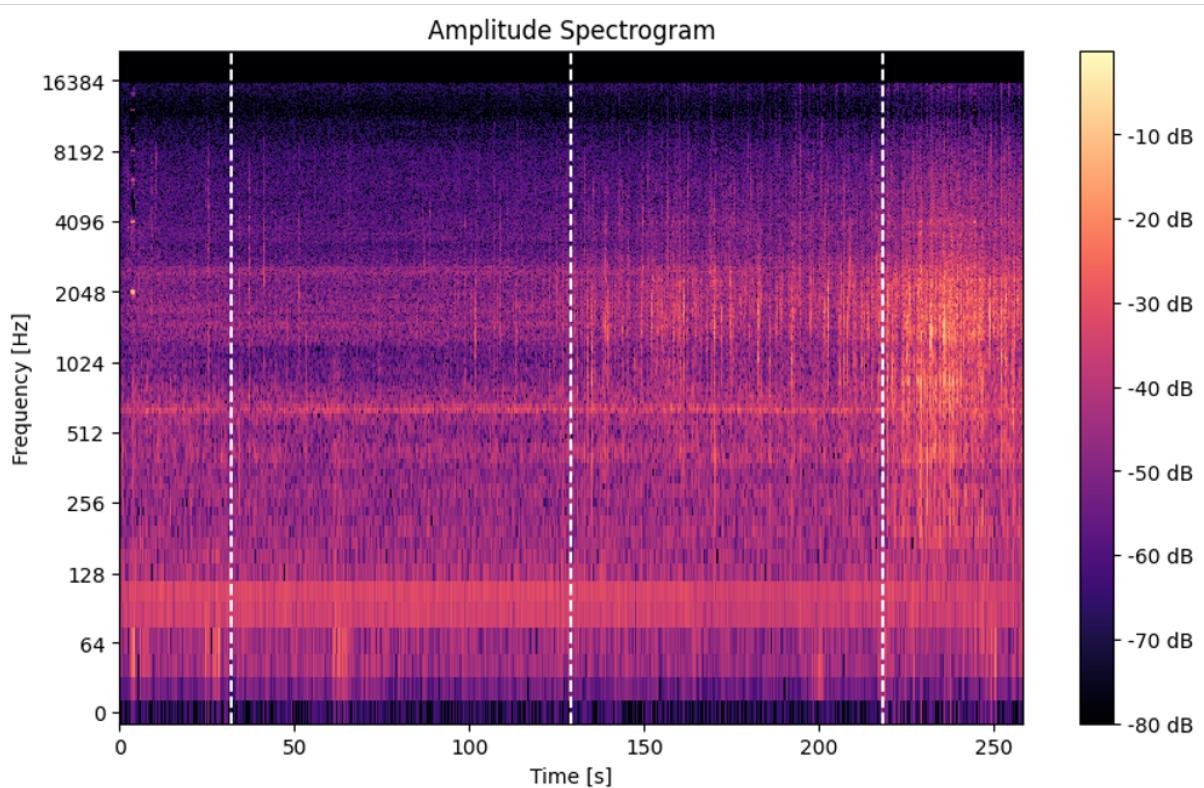


図 5.4: 調理音 3 における振幅スペクトログラムに基づくラベリング結果

このようにして、各調理音の振幅スペクトログラムを解析し、再度ラベリングした結果を表5.8に示す。

結果、2クラスでラベリングしていた時よりも、揚げ途中が細分化されたため、データセットの不均等な状態が緩和された。本研究では揚げ終わりの識別に重点を置いているため、ラベル3と揚げ終わりであるラベル4（揚げ終わり）の二値分類による性能評価を実施する。

#### 5.2.5 4クラスにおけるMFCC特徴量による分類精度の評価

前項でのラベリング結果を基に、MFCC特徴量を利用して、分類精度の評価を実施する。MFCCを抽出した際のデータバランスは、クラス1:クラス2:クラス3:揚げ終わり =

表 5.8: 振幅スペクトログラムに基づくラベリング結果一覧

	ラベル 1 [s]	ラベル 2 [s]	ラベル 3 [s]	ラベル 4 (Finish) [s]
調理音 1	0 - 45	45 - 123	123 - 214	214 - 269
調理音 2	0 - 50	50 - 129	129 - 238	238 - 280
調理音 3	0 - 32	32 - 129	129 - 218	218 - 258
調理音 4	0 - 43	43 - 147	147 - 225	225 - 278
調理音 5	0 - 60	60 - 120	120 - 176	176 - 246
調理音 6	0 - 75	75 - 174	174 - 208	208 - 300
調理音 7	0 - 34	34 - 136	136 - 195	195 - 313
調理音 8	0 - 28	28 - 100	100 - 215	215 - 361
調理音 9	0 - 43	43 - 106	106 - 157	157 - 300
調理音 10	0 - 63	63 - 98	98 - 185	185 - 270

1,812:3,156:3,076:3,434 となった。このうち、クラス 3 および揚げ終わりクラスの 2 クラスによる分類を実施する。

まず、図 5.5 に再度ラベリングされた結果を基にした PCA 結果を示す。左図は 4 クラスすべての PCA 結果をプロットしており、右図は分類対象とするクラス 3 および揚げ終わりをプロットした結果である。

4 クラスの結果より、クラス 1 およびクラス 2 は明確なクラスタを形成している部分があるが、クラス 3 および揚げ終わりとの境界は曖昧である。右図より、クラス 3 と揚げ終わりは重なる領域が多いことがわかる。また、クラスタ間に明確な境界がない。

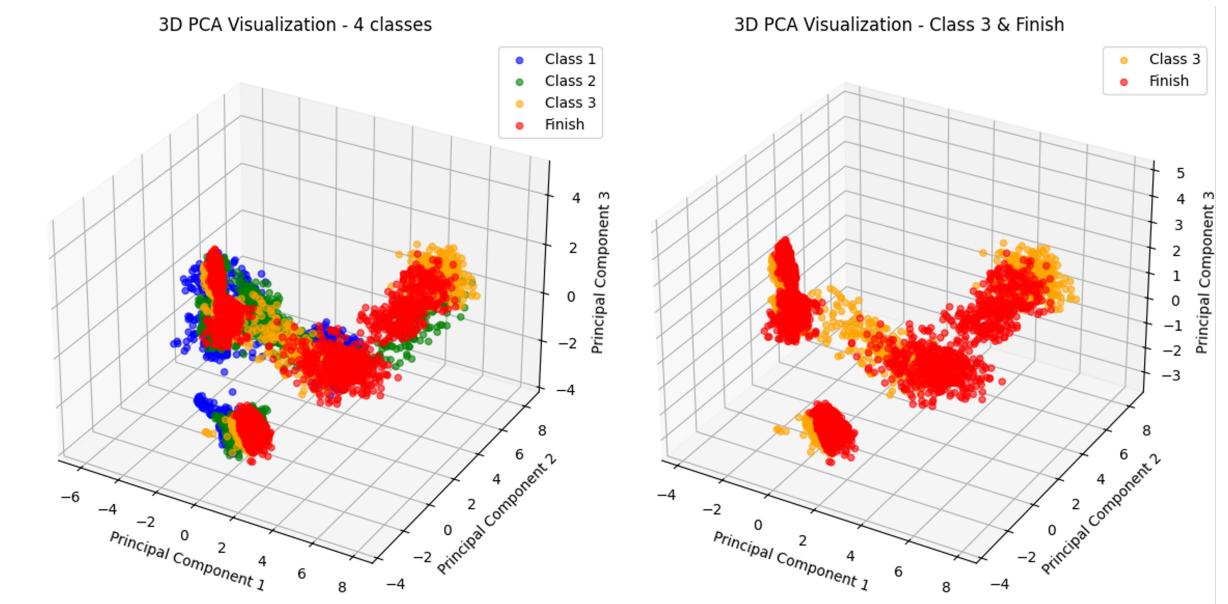


図 5.5: 4 クラスラベルにおける MFCC の PCA 結果（左図:4 クラス、右図:2 クラス）

次に、t-SNE による分析を図 5.6 に示す。左図より、クラス 1 およびクラス 2 は比較的分離している。また右図より、クラス 3 および揚げ終わりは、大部分が重なっており、完全に分離されていない。いくつかのクラスタは形成されているが、明確な境界が見られないことがわかる。

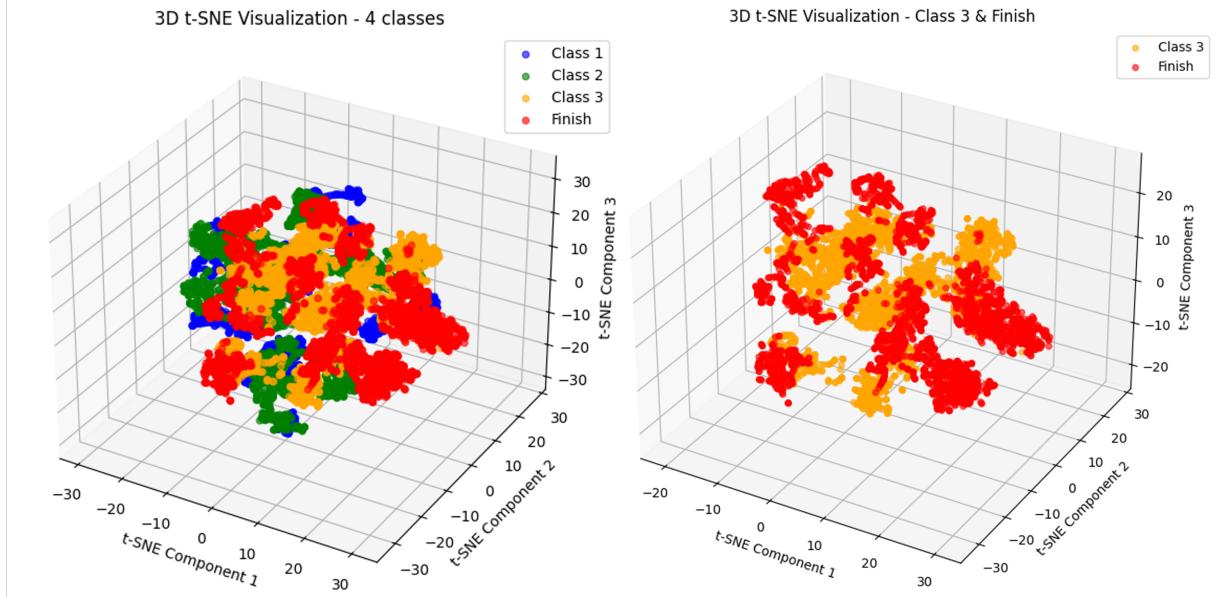


図 5.6: 4 クラスラベルにおける MFCC の t-SNE 結果 (左図:4 クラス, 右図:2 クラス)

PCA および t-SNE の解析結果より, クラス 3 および揚げ終わりは, 特徴空間において極めて似た振る舞いをしている可能性がある. また, 現在の特徴セットのみでは, 揚げ途中と揚げ終わりを十分に区別できないと考えられる.

表 5.9 に, クラス 3 および揚げ終わりによる二値分類の結果を示す. リサンプリングを適用した際の最も良い分類精度は, F1 スコアが  $57.6\% \pm 32.4\%$ , 適合率が  $72.9\% \pm 19.3\%$  であった. この精度と比較すると, 全体的に適合率は向上しているが, F1 スコアはどの分類器も上回っていない.

表 5.9: MFCC を利用した分類器ごとのクラス 3 および揚げ終わりの分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.496 \pm 0.316$	$0.649 \pm 0.282$	$0.620 \pm 0.440$	$0.587 \pm 0.117$
ランダムフォレスト	$0.509 \pm 0.318$	$0.740 \pm 0.238$	$0.598 \pm 0.409$	$0.600 \pm 0.146$
GNB	$0.471 \pm 0.413$	$0.799 \pm 0.203$	$0.588 \pm 0.507$	$0.662 \pm 0.116$
kNN	$0.409 \pm 0.294$	$0.668 \pm 0.250$	$0.455 \pm 0.388$	$0.520 \pm 0.194$
$SVM_{linear}$	$0.494 \pm 0.308$	$0.648 \pm 0.282$	$0.617 \pm 0.437$	$0.584 \pm 0.110$
$SVM_{poly}$	$0.456 \pm 0.390$	$0.790 \pm 0.217$	$0.549 \pm 0.481$	$0.609 \pm 0.181$
$SVM_{rbf}$	$0.489 \pm 0.371$	$0.796 \pm 0.221$	$0.581 \pm 0.469$	$0.620 \pm 0.161$
$SVM_{sigmoid}$	$0.463 \pm 0.378$	$0.784 \pm 0.283$	$0.562 \pm 0.456$	$0.563 \pm 0.261$
XGB	$0.426 \pm 0.279$	$0.638 \pm 0.285$	$0.455 \pm 0.372$	$0.540 \pm 0.146$

以上の結果より, クラス 3 および揚げ終わりは, 特徴空間において極めて似た振る舞いをしている可能性が高く, 現在の特徴セットのみでは, 揚げ終わりを十分に分類することができないことがわかった.

### 5.2.6 ウエーブレット変換による特徴量抽出を利用した分類精度の評価

次に、ウェーブレット変換を利用した音響特徴量の抽出を検討する。図5.7はPCA結果であり、左図は4クラス、右図はクラス3および揚げ終わりをプロットしている。左図より、各クラスは全体的に密集しており、分離していないことがわかる。また、右図より揚げ終わりはクラス3よりも広域に分布しており、一部の揚げ終わりデータは外側に散らばっていることがわかる。このことから、揚げ終わりとクラス3は異なる特徴を持つ可能性がある。しかし、両クラスともに混在している領域があるため、完全な分離は困難であると推測される。

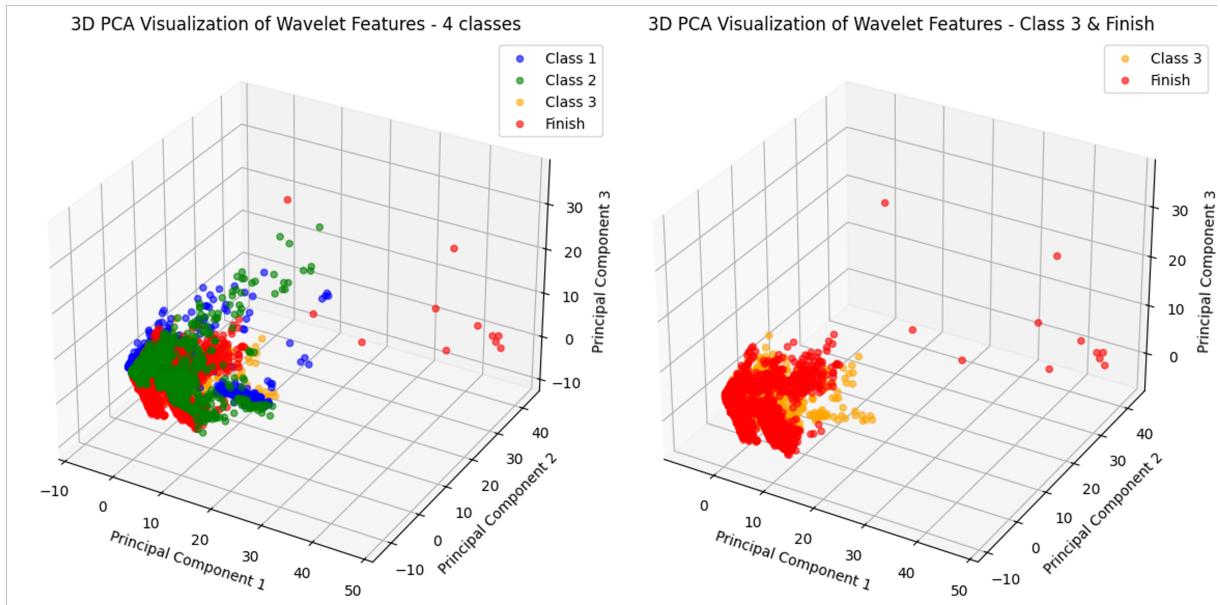


図 5.7: 4 クラスラベルにおけるウェーブレット変換から得られる特徴量のPCA結果  
(左図:4 クラス, 右図:2 クラス)

図5.8はt-SNEによる解析結果である。左図の4クラス描画結果から、PCAよりもクラス間のまとまりが明確になっていることがわかる。また、右図のクラス3および揚げ終わりのプロットより、t-SNEの結果でも両クラスは混在しており、明確な境界がないことがわかる。

以上の解析結果より、クラス3と揚げ終わりの間には明確な境界が存在しない可能性が高く、ウェーブレット変換から算出した特徴量では、クラスの識別に有効でないと考えられる。

表5.10に、クラス3および揚げ終わりによる二値分類の結果を示す。表5.9の結果と比較しても、特筆して精度の差が開いている分類器は存在しない。

以上の結果より、本研究でウェーブレット変換から算出した特徴量では、クラス3および揚げ終わりの分類が困難であることがわかった。

## 5.3 変化点検出による唐揚げの揚げ終わり検出精度の評価

変化点検出における精度評価は、揚げ途中から揚げ終わりへの変化点（真の変化点）に対する検出精度を評価するために、適合率、再現率、F1スコアを使用する。ここで適合

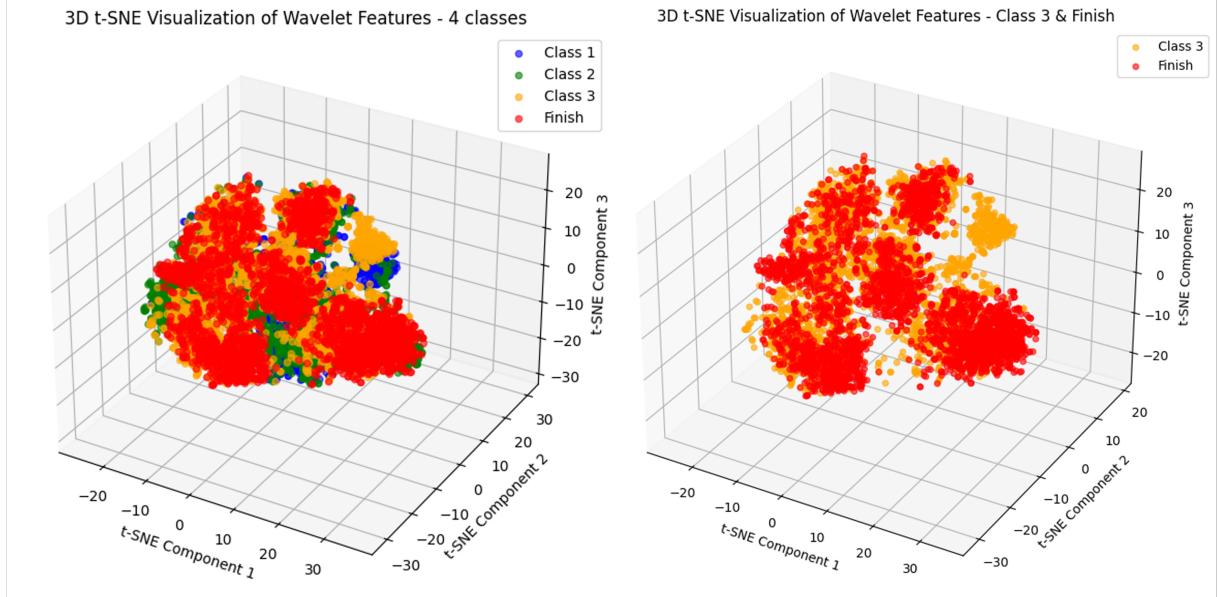


図 5.8: 4 クラスラベルにおけるウェーブレット変換から得られる特徴量の t-SNE 結果  
(左図:4 クラス, 右図:2 クラス)

率は、検出した変化点のうち、どれだけが真の変化点であったかを表す。再現率は、真の変化点がどれだけ正しく検出されたかを表す。真の変化点の検出有無は、許容範囲を設定して、その範囲内に検出点が収まっていた場合、検出されたこととする。許容範囲は、真の変化点  $\pm 10$  秒と設定する。この値は、1.1.4 で述べた通り、食中毒を防ぐための十分な加熱は、鶏肉の中心の温度が  $75^{\circ}\text{C}$  以上で 1 分間以上とされており、前節で再定義した揚げ終わりラベルの時間はすべて調理開始から 1 分以上経過していることから、危険ではない範囲であることがわかる。また、変化点検出には、時間ごとの振幅の平均を特徴として利用した。調理音 1 から調理音 4 は雑音が含まれているため、振幅の平均に対してメディアンフィルタを適用する。

解析の結果、最も揚げ終わりの変化点を捉えることが可能なパラメータは、ウィンドウ幅 30、ペナルティ 6.5 であった。以下に、このパラメータを設定した変化点検出による各調理音の検出結果を示す。

図 5.9 は調理音 1、図 5.10 は調理音 2 の変化点検出結果である。調理音 1 の真の変化点は 214 秒であり、それに最も近い変化点は 214.601 秒で検出されているため、正確に変化点を捉えている。調理音 2 の真の変化点は 238 秒、それに最も近い変化点は 259.383 秒であった。調理音 2 に対しては正確に変化点を捉えられなかったことがわかる。

図 5.11 は調理音 3、図 5.12 は調理音 4 の変化点検出結果である。調理音 3 の真の変化点は 218 秒であり、検出された変化点は 219.413 秒であった。また、調理音 4 は真の変化点が 225 秒、検出された変化点は 226.778 秒であった。どちらの変化点検出においても、真の変化点と 2 秒以内の誤差であり、他の変化点も検出されていない。

図 5.13 は調理音 5、図 5.14 は調理音 6 の変化点検出結果である。調理音 5 の真の変化点は 176 秒であり、それに最も近い変化点は 176.851 秒であった。また、調理音 6 は真の変化点が 208 秒、それに最も近い変化点は 206.891 秒であった。どちらも誤差が 2 秒以内に収まっているが、他の変化点も検出されている。

表 5.10: ウェーブレット変換を利用した分類器ごとのクラス 3 および揚げ終わりの分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.550 \pm 0.256$	$0.668 \pm 0.251$	$0.583 \pm 0.347$	$0.611 \pm 0.168$
ランダムフォレスト	$0.419 \pm 0.339$	$0.493 \pm 0.371$	$0.464 \pm 0.419$	$0.542 \pm 0.203$
GNB	$0.447 \pm 0.359$	$0.779 \pm 0.266$	$0.445 \pm 0.412$	$0.607 \pm 0.217$
kNN	$0.439 \pm 0.246$	$0.574 \pm 0.325$	$0.525 \pm 0.345$	$0.445 \pm 0.212$
$SVC_{linear}$	$0.529 \pm 0.250$	$0.664 \pm 0.256$	$0.554 \pm 0.347$	$0.591 \pm 0.167$
$SVC_{poly}$	$0.226 \pm 0.337$	$0.734 \pm 0.350$	$0.300 \pm 0.460$	$0.503 \pm 0.184$
$SVC_{rbf}$	$0.349 \pm 0.315$	$0.674 \pm 0.330$	$0.415 \pm 0.425$	$0.493 \pm 0.201$
$SVC_{sigmoid}$	$0.449 \pm 0.207$	$0.643 \pm 0.263$	$0.502 \pm 0.316$	$0.487 \pm 0.159$
XGB	$0.430 \pm 0.339$	$0.510 \pm 0.373$	$0.485 \pm 0.436$	$0.541 \pm 0.214$

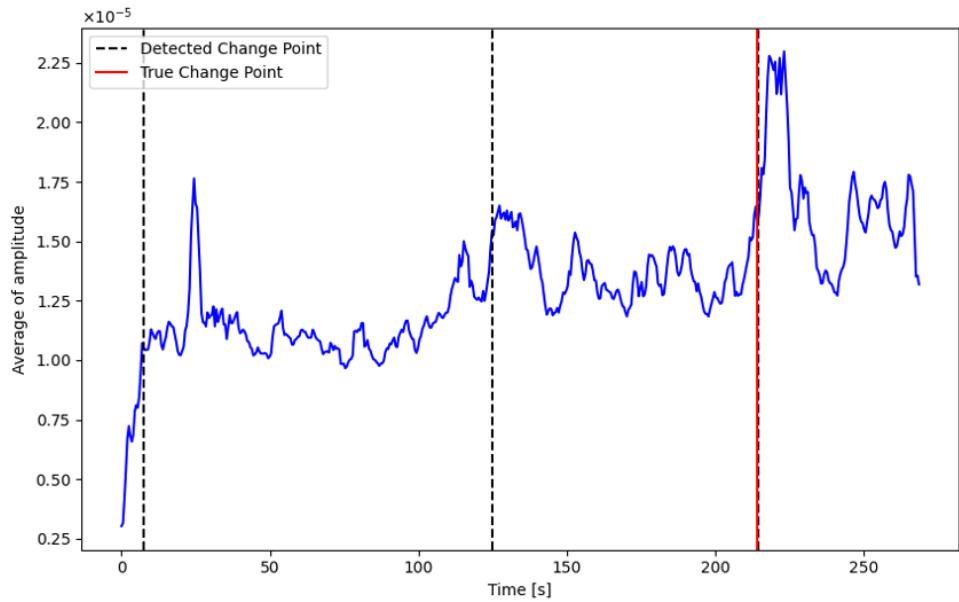


図 5.9: 調理音 1 の変化点検出結果

図 5.15 は調理音 7, 図 5.16 は調理音 8 の変化点検出結果である。調理音 7 の真の変化点は 195 秒であり, それに最も近い変化点は 192.168 秒であった。また, 調理音 8 は真の変化点が 215 秒, それに最も近い変化点は 212.159 秒であった。どちらも誤差が 4 秒以内に収まっているが, 調理音 5, 6 と同様に他の変化点も検出されている。

図 5.17 は調理音 9, 図 5.18 は調理音 10 の変化点検出結果である。調理音 9 の真の変化点は 157 秒であり, それに最も近い変化点は 157.208 秒であった。また, 調理音 10 は真の変化点は 185 秒, それに最も近い変化点は 181.882 秒であった。調理音 9 については, 誤差がほとんどないことがわかる。調理音 10 は誤差が 4 秒以内に収まっている。また, 両結果ともに他の変化点も検出されている。

調理音 1 から 10 までの変化点検出の評価一覧および, 真の変化点と検出された最も近い変化点を表 5.11 に示す。調理音 2 以外は, 真の変化点を正しく検出していることがわかる。また, 各調理音の変化点検出結果の図より, 150 秒以降に初めて検出された変化点

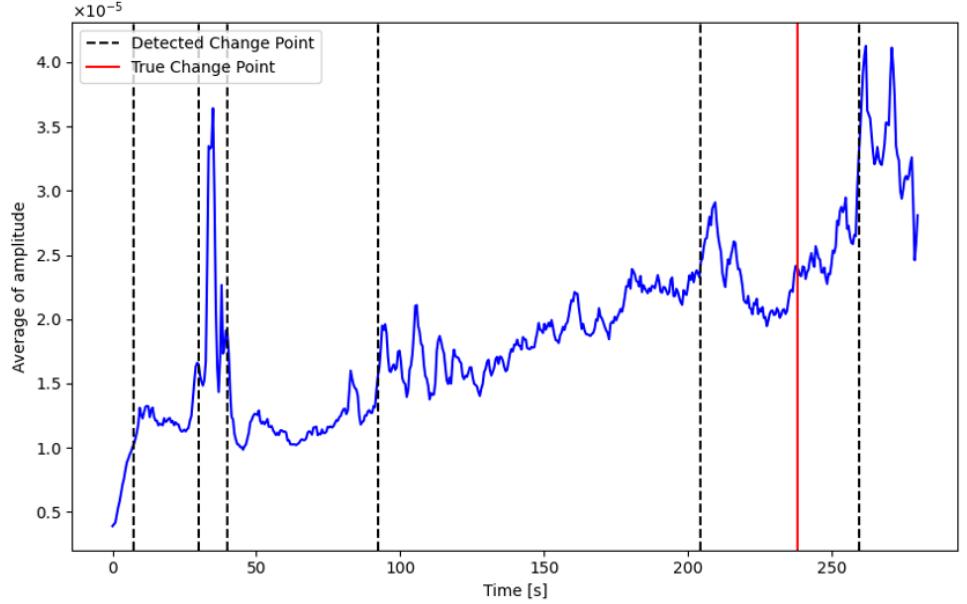


図 5.10: 調理音 2 の変化点検出結果

が真の変化点であることがわかる。150秒以降と限定することで、調理音 2 以外は真の変化点が正確に検出可能であり、誤検出も調理音 2 以外は存在しない。つまり、この制約により、揚げ終わりの検出精度は 3 種類すべての指標において 90% となり、適合率が 90% であることから、誤検出の可能性は 10% となる。

これに対して、機械学習手法はモデルの精度が安定していない。よって、本研究においては、機械学習手法よりも変化点検出の方が、揚げ終わり識別に対して精度が高いと言える。このことから、変化点検出を利用した唐揚げの調理音識別が可能であると考え、FCGs には変化点検出アルゴリズムを実装する。

表 5.11: 各調理音の変化点および精度一覧

	F1 スコア	適合率	再現率	真の変化点 [s]	検出された変化点 [s]
調理音 1	0.500	0.333	1.000	214	214.601
調理音 2	0.000	0.000	0.000	238	259.383
調理音 3	1.000	1.000	1.000	218	219.413
調理音 4	1.000	1.000	1.000	225	226.778
調理音 5	0.400	0.250	1.000	176	176.851
調理音 6	0.667	0.500	1.000	208	206.891
調理音 7	0.500	0.333	1.000	195	192.168
調理音 8	0.500	0.333	1.000	215	212.159
調理音 9	0.333	0.200	1.000	157	157.208
調理音 10	0.500	0.333	1.000	185	181.882

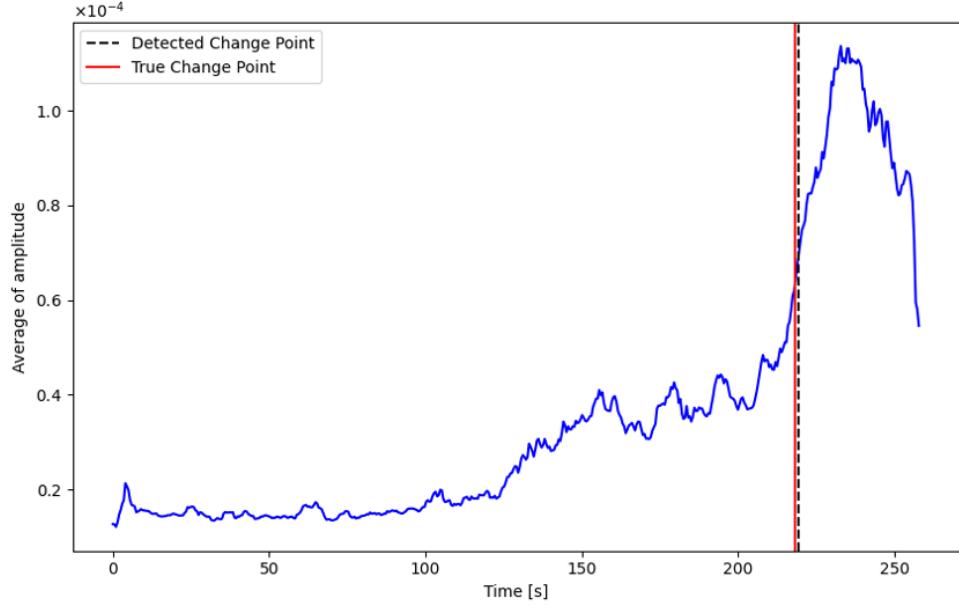


図 5.11: 調理音 3 の変化点検出結果

## 5.4 変化点検出を利用した揚げ終わり判定システムの実装

前節までの結果より、揚げ終わり識別手法として変化点検出を利用する。スマートフォンアプリケーションの実装は、Android Studio を使用した [73]。また、変化点検出のアルゴリズムは Chaquopy を利用して実装した [74]。Chaquopy は Android アプリに Python コンポーネントを組み込み可能とする SDK である。

録音を開始してから、揚げ終わりを判定するまでのシステムフローチャートを図 5.19 に示す。

変化点検出は録音を開始してから 165 秒後から開始する。これは、150 秒地点が変化点検出のウィンドウの中心となるようにするためである。変化点検出に利用するデータ量は 30 秒分であり、165 秒経過した段階で特徴量を算出する。特徴量は振幅スペクトログラムを算出するため、STFT 適用後、振幅スペクトルを求める。その後、メディアンフィルタで雑音を低減し、時間ごとのエネルギーを算出する。そのエネルギーを基に変化点を検出する。検出された場合は、図 3.1 のような画面によるフィードバックを実施する。これにより、アプリケーション利用者は、唐揚げが揚がったタイミングを認知する。

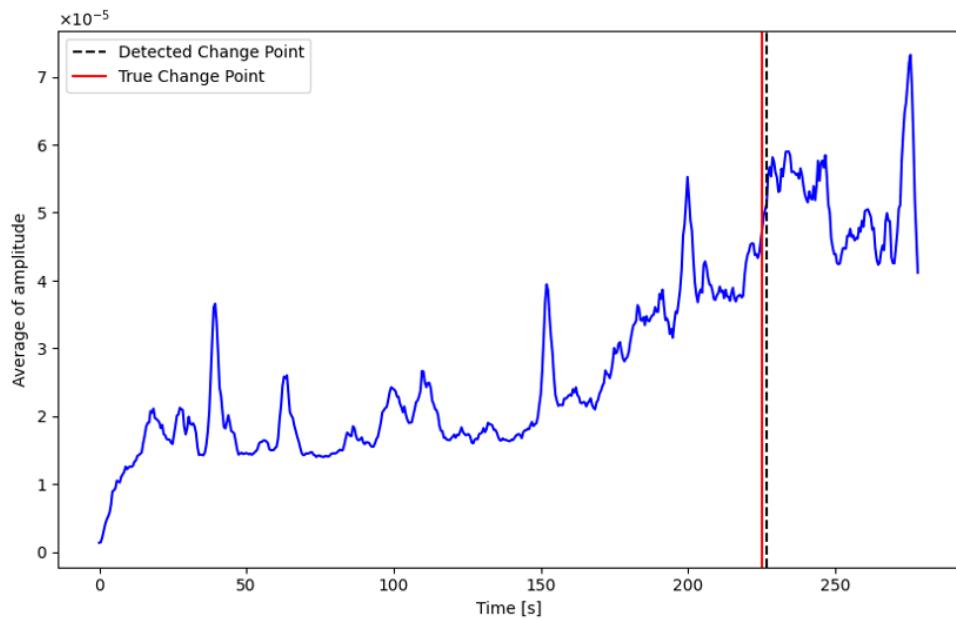


図 5.12: 調理音 4 の変化点検出結果

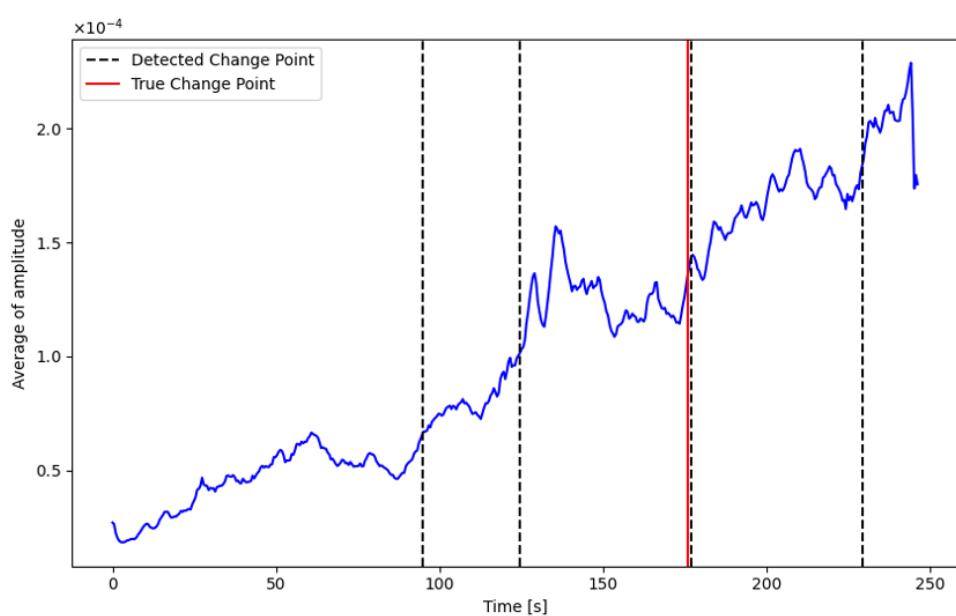


図 5.13: 調理音 5 の変化点検出結果

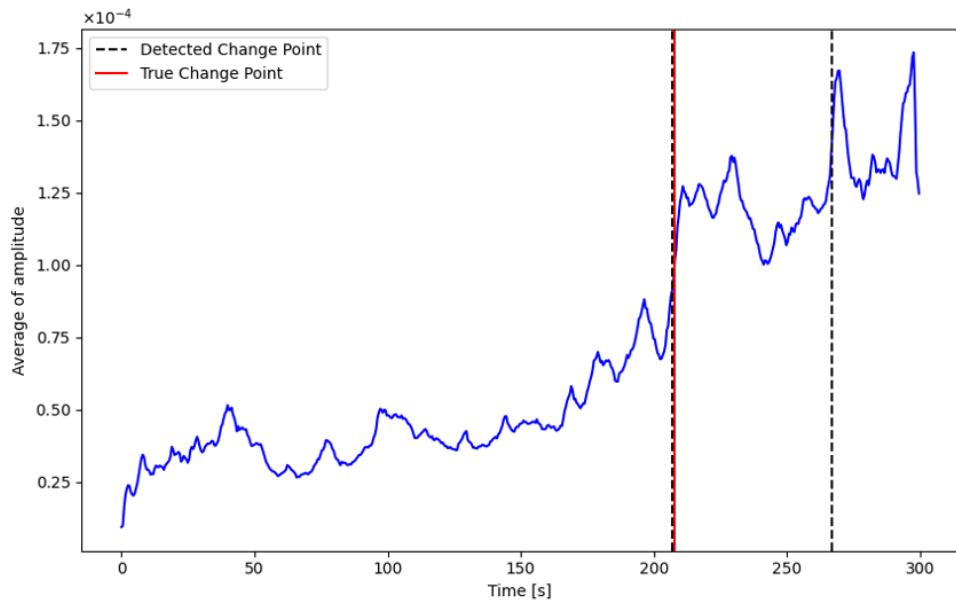


図 5.14: 調理音 6 の変化点検出結果

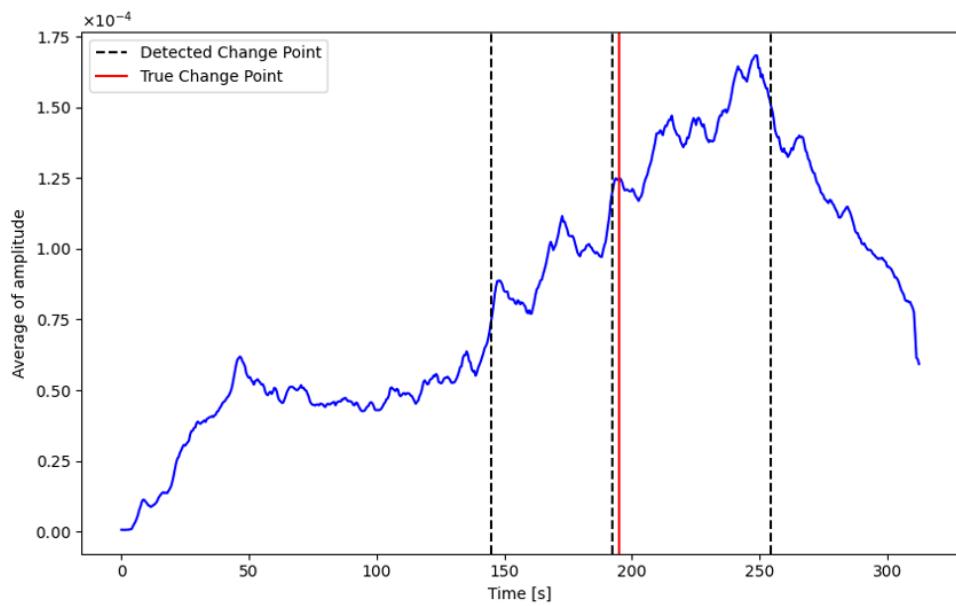


図 5.15: 調理音 7 の変化点検出結果

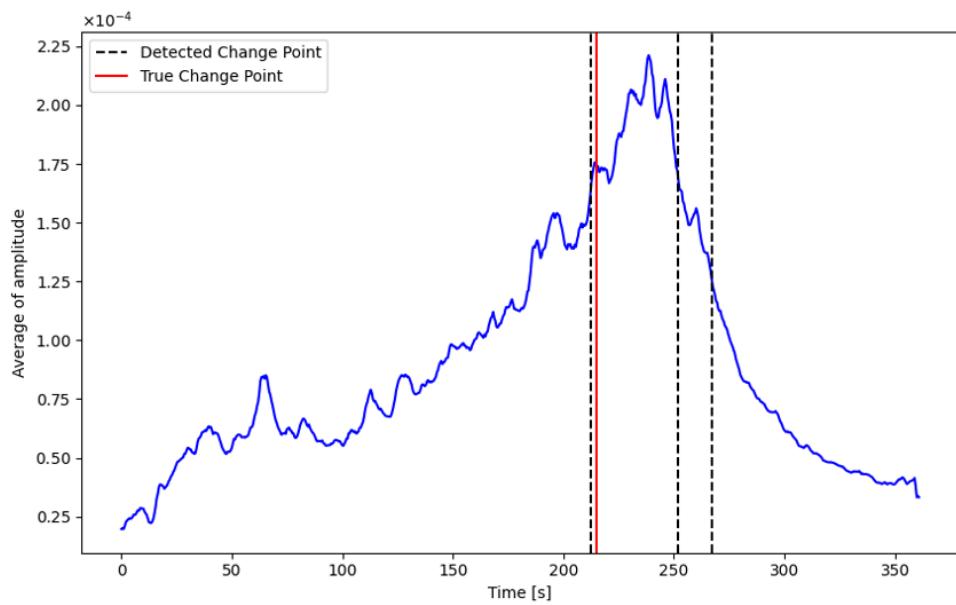


図 5.16: 調理音 8 の変化点検出結果

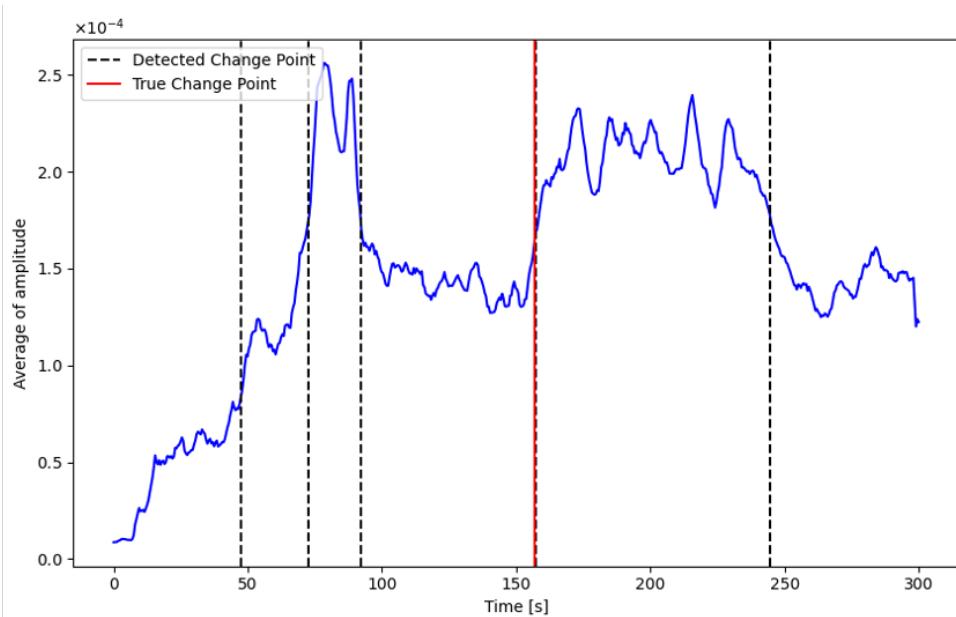


図 5.17: 調理音 9 の変化点検出結果

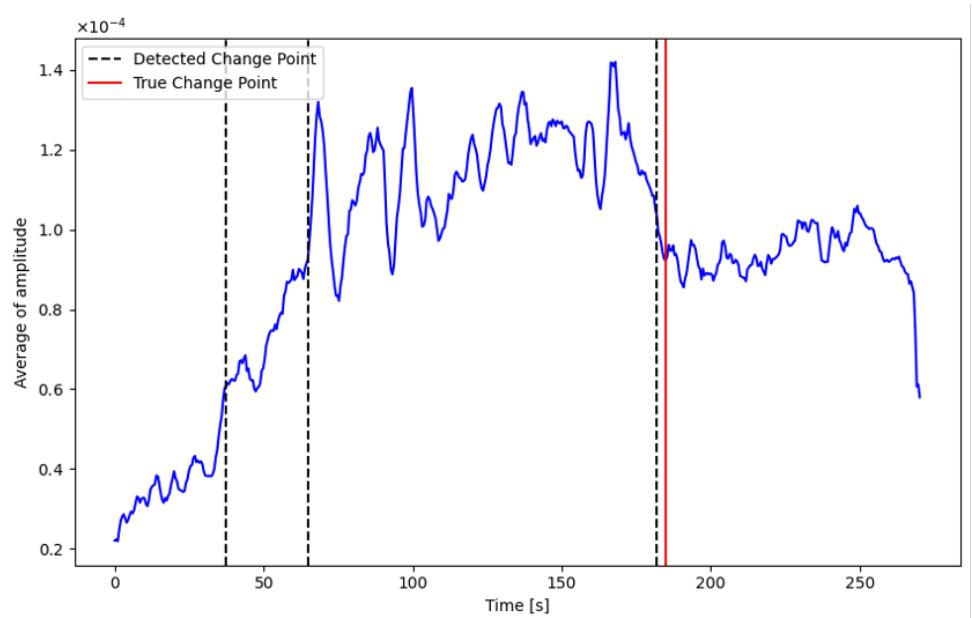


図 5.18: 調理音 10 の変化点検出結果

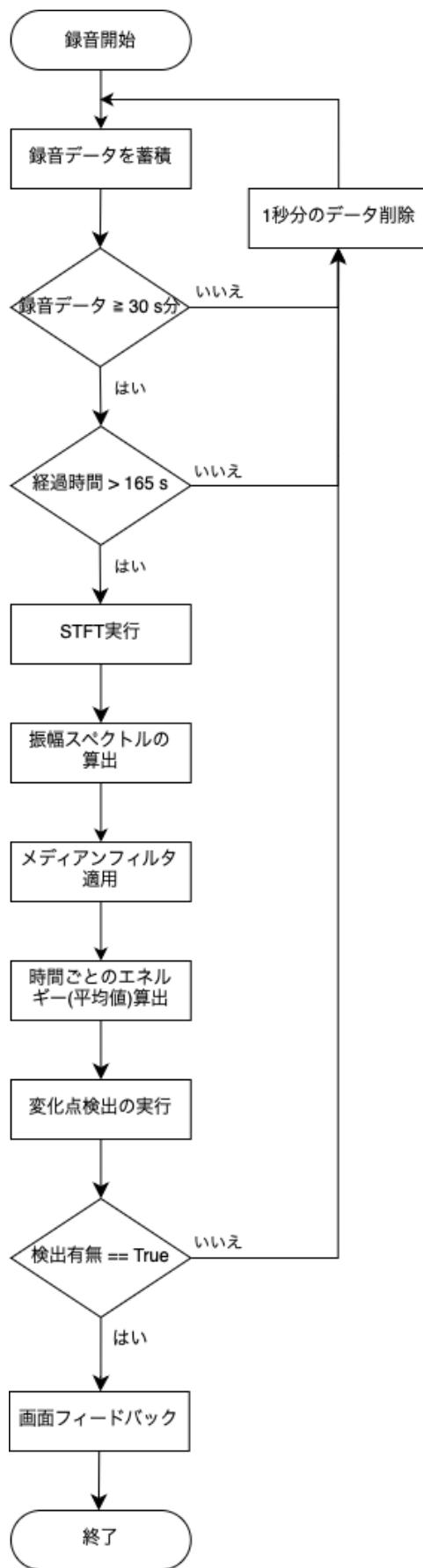


図 5.19: FCGs のシステムフローチャート

# 第6章 結論

本章では、本研究のまとめおよび今後の展望について述べる。

## 6.1 本研究のまとめ

本研究では、唐揚げの揚げ終わりの見極めが料理初心者にとって課題となっていることに対して、料理中でも容易に利用可能なスマートフォン向け調理支援システム FCGs を開発した。

また、揚げ終わりとなる前の状態で調理完了と誤判定しない安全な判定システムを構築するために、機械学習および変化点検出手法を検討した。機械学習手法においては、リサンプリング手法を組み合わせて、分類精度の向上を図った。また音響特徴量として、MFCC、ウェーブレット変換を検討した。変化点検出手法はWindowベースのアルゴリズムを検討した結果、機械学習手法よりも高い精度で、揚げ終わりの検出が可能であることが判明した。よって、FCGsには、変化点検出による揚げ終わり検出を実装し、調理支援システムを開発した。

本研究の目標に対する結果としては以下の通りである。

1. 唐揚げの調理状態において、揚げ終わりとなる前の状態で調理完了と誤判定しない安全な音判定システムの構築を目指した。結果、変化点検出手法を利用することで、揚げ終わりの検出における再現率が 90% を達成した。また、調理開始から 150 秒以降の変化点を揚げ終わりの変化点として検出することで、F1 スコア、適合率、再現率の 3 種類の指標において 90% の精度を達成し、誤検出は 10% となった。
2. スマートフォン上で唐揚げの揚げ終わりをリアルタイムに判別する、容易に利用可能なアプリケーションの開発を目指した。結果、変化点検出アルゴリズムを利用して、スマートフォンのマイクロフォンからリアルタイムにデータを取得し、唐揚げの揚げ終わりを判定するアプリケーションを開発した。

## 6.2 今後の展望

今後の展望として、揚げ終わり識別手法の精度向上が挙げられる。本研究で検討した機械学習手法は、調理音データの時系列情報を考慮していなかったため、LSTM といった時系列を考慮した機械学習モデルを利用することで、分類精度の向上が可能であると考える [75]。本研究でも LSTM を検証したが、時系列データが不足していたため、十分な結果を得ることができず断念した。よって、データセットの拡充を実施して、より頑健な揚げ終わり判定システムの構築を検討したい。

また、調理油の温度および調理環境の違いにも対応可能なアルゴリズムおよび分類モデルの構築を目指す。料理初心者は、油の温度を正確に測定することが困難であると考えら

れる。よって、油の温度がある程度、異なっていても問題なく調理可能な手法を検討する必要があると考える。調理環境においても、騒音の種類や使用する調理器具の違いなどにより、音響が異なってくるため、調理環境の違いにも対応可能な手法が必要である。

本研究では、唐揚げの調理が完了したタイミングを検出する調理支援にとどまっている。そのため、検出されたタイミングが最も美味しい状態であるかは判別できていない。しかし、クリスピ一度に関する研究は、これまで数多く実施されている[76][77][78][79]。クリスピ一度は唐揚げの食感として表現される感覚である。よって、美味しさに対する最適なタイミングを検知し、美味しい唐揚げを調理する支援システムに拡張可能であると考える。

精度向上の手法として、画像分析も利用可能である。スマートグラスのようなマイクロフォンおよびカメラを搭載したデバイス利用することで、マルチモーダルシステムによる精度向上が期待される。また、フィードバックをスマートグラス上に表示可能であるため、ユーザビリティの高い調理支援システムの開発が可能であると考える。

また、本研究の最終的な目標として、揚げ物料理すべてに対応可能な調理支援システムを開発を考えている。そのため、唐揚げだけでなく、天ぷら、コロッケ、豚カツ、白身魚のフライなどのレシピに対しても、研究を実施したいと考えている。その結果、料理初心者の料理へのハードルを下げ、自炊を習慣付けるモチベーションにつながることを期待する。

## 謝辞

本研究を進めるにあたって、学部4年から3年間、ご指導いただいた青山学院大学理工学部情報テクノロジー学科 Guillaume Lopez 教授に深く感謝申し上げます。研究方針、研究における課題、就職活動に至るまでご指導いただいたこと、特に、毎週の研究会で、毎度的確なご助言をいただいたこと、心から感謝しております。加えて、私の力不足により、機械学習の分類精度が中々良くならず、学会発表や論文誌への投稿が叶わなかったこと、お詫び申し上げます。

また、大学関連の事務手続きおよび研究室の整備等、お心遣いいただいた研究補助員の大熊ちひろ様にも深く感謝申し上げます。

さらに、副査として、残り時間で実現可能な識別手法について的確なご助言をいただきました青山学院大学理工学部情報テクノロジー学科大原剛三教授に深く感謝申し上げます。

そして、研究生活において日々研究の助言をいただいたウェアラブル環境情報システム研究室の同期の皆様、並びに学部4年時にデータ収集にご協力いただいただけでなく、大学生活を支えてくださった母にも深く感謝申し上げます。

2025年1月31日  
新井 優作

## 参考文献

- [1] 日本政策金庫公庫. 飲食店のテイクアウト・デリバリー等に関する消費者調査結果. [https://www.jfc.go.jp/n/findings/pdf/seikatsu20\\_1215a.pdf](https://www.jfc.go.jp/n/findings/pdf/seikatsu20_1215a.pdf), 12 2020. (最終参照日 : 2024/12/31).
- [2] 日本政策金庫公庫. 中食と外食に関する消費者動向調査結果. [https://www.jfc.go.jp/n/findings/pdf/topics\\_180913a.pdf](https://www.jfc.go.jp/n/findings/pdf/topics_180913a.pdf), 9 2018. (最終参照日 : 2024/12/31).
- [3] 農林水産省. 新型コロナウイルス感染症の拡大による食生活の変化. <https://www.maff.go.jp/j/syokuiku/ishiki/r03/zuhyou/z9-1.html>, 3 2021. (最終参照日 : 2024/12/31).
- [4] 日本政策金庫公庫. 食の志向等に関する調査結果. [https://www.jfc.go.jp/n/findings/pdf/topics\\_200805a.pdf](https://www.jfc.go.jp/n/findings/pdf/topics_200805a.pdf), 7 2020. (最終参照日 : 2024/12/31).
- [5] Celia Rodríguez-Pérez, Esther Molina-Montes, Vito Verardo, Reyes Artacho, Belén García-Villanova, Eduardo Jesús Guerra-Hernández, and María Dolores Ruíz-López. Changes in dietary behaviours during the covid-19 outbreak confinement in the spanish covidiet study. *Nutrients*, Vol. 12, No. 6, p. 1730, 2020.
- [6] Danijela Pfeifer, Josip Rešetar, Jasenka Gajdoš Kljusurić, Ines Panjkota Krbavčić, Darija Vranešić Bender, Celia Rodríguez-Pérez, María Dolores Ruíz-López, and Zvonimir Šatalić. Cooking at home and adherence to the mediterranean diet during the covid-19 confinement: the experience from the croatian covidiet study. *Frontiers in nutrition*, Vol. 8, p. 617721, 2021.
- [7] Sarah Gerritsen, Victoria Egli, Rajshri Roy, Jill Haszard, Charlotte De Backer, Lauranna Teunissen, Isabelle Cuykx, Paulien Decorte, Sara Pabian Pabian, Kathleen Van Royen, et al. Seven weeks of home-cooked meals: Changes to new zealanders' grocery shopping, cooking and eating during the covid-19 lockdown. *Journal of the Royal Society of New Zealand*, Vol. 51, No. sup1, pp. S4–S22, 2021.
- [8] ハウス食品株式会社. さっと…レシピの独特表現「レシピ語」に料理初心者の約8割が戸惑っていることが判明! [https://housefoods.jp/company/news/pdf/20210517\\_release\\_v2.pdf](https://housefoods.jp/company/news/pdf/20210517_release_v2.pdf), 5 2021. (最終参照日 : 2025/1/2).
- [9] 平島円, 磯部由香, 堀光代. 学生の「切り方」に対する認知度と自信度の変化. 日本調理科学会誌, Vol. 54, No. 5, pp. 234–243, 2021.
- [10] Yuta Kido, Teruhiro Mizumoto, Hirohiko Suwa, Yutaka Arakawa, and Keiichi Yamamoto. A cooking support system for seasoning with smart cruet. In *International Conference on Human-Computer Interaction*, pp. 369–382. Springer, 2019.

- [11] 加藤岳大, 横窪安奈, ロペズギヨーム. Syncook: 動画メタデータと加速度センサを用いたレシピ動画進行度自動同期システム. <https://www.wiss.org/WISS2020Proceedings/data/N-15.pdf>, 2020. (最終参照日 : 2025/1/2).
- [12] 株式会社 Felicidad. Hestan cue ヘスタンキューフォード公式サイト. <https://hestancue.jp/>. (最終参照日 : 2025/1/2).
- [13] KY Voong, AB Norton, TB Mills, and IT Norton. Characterisation of deep-fried batter and breaded coatings. *Food Structure*, Vol. 16, pp. 43–49, 2018.
- [14] Franco Pedreschi, Pamela Hernández, Clara Figueroa, and Pedro Moyano. Modeling water loss during frying of potato slices. *International journal of food properties*, Vol. 8, No. 2, pp. 289–299, 2005.
- [15] 国立感染症研究所. サルモネラ感染症とは. <https://www.niid.go.jp/niid/ja/kansennohanashi/409-salmonella.html>. (最終参照日 : 2024/12/11).
- [16] 国立感染症研究所. カンピロバクター感染症とは. <https://www.niid.go.jp/niid/ja/kansennohanashi/385-campylobacter-intro.html>. (最終参照日 : 2024/12/11).
- [17] 厚生労働省. カンピロバクター食中毒予防について (q&a) . [https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/kenkou\\_iryou/shokuhin/syokuchu/campylobacterqa.html](https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/kenkou_iryou/shokuhin/syokuchu/campylobacterqa.html). (最終参照日 : 2024/12/11).
- [18] 食品安全委員会. カンピロバクターによる食中毒にご注意ください. [https://www.fsc.go.jp/sonota/e1\\_campylo\\_chudoku\\_20160205.html](https://www.fsc.go.jp/sonota/e1_campylo_chudoku_20160205.html). (最終参照日 : 2024/12/11).
- [19] 農林水産省. 鶏料理を楽しむために～カンピロバクターによる食中毒にご注意を !! ~. <https://www.kurashikagaku.co.jp/report/detail.php?id=133>. (最終参照日 : 2024/12/11).
- [20] Solveig Langsrud, Oddvin Sørheim, Silje Elisabeth Skuland, Valérie Lengard Almli, Merete Rusås Jensen, Magnhild Seim Grøvlen, Øydis Ueland, and Trond Møretrø. Cooking chicken at home: Common or recommended approaches to judge doneness may not assure sufficient inactivation of pathogens. *PLoS One*, Vol. 15, No. 4, p. e0230928, 2020.
- [21] engadget. New android feature alerts you to smoke alarms and other 'critical' sounds. <https://www.engadget.com/android-sound-notification-alerts-170045534.html>, 10 2020. (最終参照日 : 2025/1/2).
- [22] Apple Inc. iphone でサウンドを認識する-apple サポート (日本). <https://support.apple.com/ja-jp/guide/iphone/iphf2dc33312/ios>. (最終参照日 : 2025/1/2).
- [23] 株式会社野村総合研究所. 第 301 回 nri メディアフォーラム 「it ナビゲーター 2021 年度版」. <https://www.nri.com/-/media/Corporate/jp/Files/PDF/knowledge/report/cc/mediaforum/2020/forum301.pdf>, 12 2020. (最終参照日 : 2025/1/2).

- [24] M Tabacchi, C Asensio, I Pavón, M Recuero, J Mir, and MC Artal. A statistical pattern recognition approach for the classification of cooking stages. the boiling water case. *Applied acoustics*, Vol. 74, No. 8, pp. 1022–1032, 2013.
- [25] 宮澤要二, 斎藤大輔, 峯松信明ほか. レシピ情報に基づく調理行動認識における音響識別器の中間特微量利用に関する検討. 研究報告音楽情報科学 (MUS), Vol. 2020, No. 31, pp. 1–6, 2020.
- [26] 陳維, 塚田紘也, 横窪安奈ほか. 音解析による唐揚げの揚がり具合判定手法. マルチメディア, 分散, 協調とモバイルシンポジウム 2022 論文集, Vol. 2022, pp. 265–270, 2022.
- [27] June Anne Caladcad, Shiela Cabahug, Mary Rose Catamco, Paul Elyson Villaceran, Leizel Cosgafa, Karl Norbert Cabizares, Marfe Hermosilla, et al. Determining philippine coconut maturity level using machine learning algorithms based on acoustic signal. *Computers and electronics in agriculture*, Vol. 172, p. 105327, 2020.
- [28] Kang Zhao, He Li, Zhihua Zha, Mingcan Zhai, and Jie Wu. Detection of sub-healthy apples with moldy core using deep-shallow learning for vibro-acoustic multi-domain features. *Measurement: Food*, Vol. 8, p. 100068, 2022.
- [29] Rafael Z Lopes and Gustavo C Dacanal. Classification of crispness of food materials by deep neural networks. *Journal of Texture Studies*, Vol. 54, No. 6, pp. 845–859, 2023.
- [30] Rui Xiao, Qunfang Hu, and Jie Li. Leak detection of gas pipelines using acoustic signals based on wavelet transform and support vector machine. *Measurement*, Vol. 146, pp. 479–489, 2019.
- [31] Anjali Malviya, Rahul Dixit, Anupam Shukla, and Nagendra Kushwaha. A novel approach to detection of covid-19 and other respiratory diseases using autoencoder and lstm. *SN Computer Science*, Vol. 6, No. 1, pp. 1–11, 2025.
- [32] Chern-Sheng Lin, Yu-Ching Pan, Yu-Xin Kuo, Ching-Kun Chen, and Chuen-Lin Tien. A study of automatic judgment of food color and cooking conditions with artificial intelligence technology. *Processes*, Vol. 9, No. 7, p. 1128, 2021.
- [33] Chao Dong, Luelue Huang, Cheng Xiong, Mengkun Li, and Jiamei Tang. Evaluation of quality of baguette bread using image analysis technique. *Journal of Food Composition and Analysis*, p. 107222, 2025.
- [34] Arman Arefi, Oliver Hensel, and Barbara Sturm. Intelligent potato frying: time to say goodbye to the “good old” processing strategies. *Thermal Science and Engineering Progress*, Vol. 34, p. 101389, 2022.
- [35] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, Vol. 2, No. 4, pp. 433–459, 2010.

- [36] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, Vol. 9, No. 11, 2008.
- [37] Leon Cohen. *Time-frequency analysis*, Vol. 778. Prentice Hall PTR New Jersey, 1995.
- [38] Ian Naismith Sneddon. *Fourier transforms*. Courier Corporation, 1995.
- [39] Fang Zheng, Guoliang Zhang, and Zhanjiang Song. Comparison of different implementations of mfcc. *Journal of Computer science and Technology*, Vol. 16, No. 6, pp. 582–589, 2001.
- [40] Stanley Smith Stevens, John Volkmann, and Edwin Broomell Newman. A scale for the measurement of the psychological magnitude pitch. *The journal of the acoustical society of america*, Vol. 8, No. 3, pp. 185–190, 1937.
- [41] Ingrid Daubechies. Ten lectures on wavelets. *Society for industrial and applied mathematics*, 1992.
- [42] Ki-Bok Kim, David K Hsu, and Daniel J Barnard. Estimation of porosity content of composite materials by applying discrete wavelet transform to ultrasonic backscattered signal. *NDT & E International*, Vol. 56, pp. 10–16, 2013.
- [43] Gilbert Strang. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.
- [44] Ingrid Daubechies and Bruce J Bates. Ten lectures on wavelets, 1993.
- [45] scikit learn. scikit-learn. <https://scikit-learn.org/stable/index.html>. (最終参照日：2025/1/21).
- [46] xgboost developers. Xgboost documentation. <https://xgboost.readthedocs.io/en/stable/>. (最終参照日：2025/1/22).
- [47] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.
- [48] Mariette Awad, Rahul Khanna, Mariette Awad, and Rahul Khanna. Support vector machines for classification. *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*, pp. 39–66, 2015.
- [49] Jun Han and Claudio Moraga. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International workshop on artificial neural networks*, pp. 195–201. Springer, 1995.
- [50] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, Vol. 4. Springer, 2006.
- [51] Thomas Bayes. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. *Philosophical transactions of the Royal Society of London*, No. 53, pp. 370–418, 1763.

- [52] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, Vol. 13, No. 1, pp. 21–27, 1967.
- [53] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 20, No. 8, pp. 832–844, 1998.
- [54] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 785–794, 2016.
- [55] Trevor Hastie, Robert Tibshirani, Jerome Friedman, Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Boosting and additive trees. *The elements of statistical learning: data mining, inference, and prediction*, pp. 337–387, 2009.
- [56] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. John Wiley & Sons, 2013.
- [57] The imbalanced-learn developers. imbalanced-learn documentation. <https://imbalanced-learn.org/stable/>. (最終参照日：2025/1/21).
- [58] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pp. 1322–1328. Ieee, 2008.
- [59] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, Vol. 16, pp. 321–357, 2002.
- [60] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pp. 878–887. Springer, 2005.
- [61] Hien M Nguyen, Eric W Cooper, and Katsuari Kamei. Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms*, Vol. 3, No. 1, pp. 4–21, 2011.
- [62] Wei-Chao Lin, Chih-Fong Tsai, Ya-Han Hu, and Jing-Shang Jhang. Clustering-based undersampling in class-imbalanced data. *Information Sciences*, Vol. 409, pp. 17–26, 2017.
- [63] Peter Hart. The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory*, Vol. 14, No. 3, pp. 515–516, 1968.
- [64] Inderjeet Mani and I Zhang. knn approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets*, Vol. 126, pp. 1–7. ICML, 2003.

- [65] Jorma Laurikkala. Improving identification of difficult small classes by balancing class distribution. In *Artificial Intelligence in Medicine: 8th Conference on Artificial Intelligence in Medicine in Europe, AIME 2001 Cascais, Portugal, July 1–4, 2001, Proceedings 8*, pp. 63–66. Springer, 2001.
- [66] Miroslav Kubat, Stan Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, Vol. 97, p. 179. Citeseer, 1997.
- [67] Tomek Ivan. Two modifications of cnn. *IEEE transactions on Systems, Man and Communications, SMC*, Vol. 6, pp. 769–772, 1976.
- [68] ruptures. Welcome to ruptures. <https://centre-borelli.github.io/ruptures-docs/>. (最終参照日：2025/1/22).
- [69] Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, Vol. 167, p. 107299, 2020.
- [70] Ioannis Pitas and Anastasios N Venetsanopoulos. *Nonlinear digital filters: principles and applications*, Vol. 84. Springer Science & Business Media, 2013.
- [71] Daniel Berrar, et al. Cross-validation., 2019.
- [72] Elan — the language archive. <https://archive.mpi.nl/tla/elan>. (最終参照日：2025/1/19).
- [73] Google LLC. Android studio. <https://developer.android.com/studio?hl=ja>. (最終参照日：2025/1/23).
- [74] Chaquo Ltd. Chaquopy python sdk for android. <https://chaquo.com/chaquopy/>. (最終参照日：2025/1/23).
- [75] S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.
- [76] Liara S Dias-Faceto, Ana Salvador, and Ana C Conti-Silva. Acoustic settings combination as a sensory crispness indicator of dry crispy food. *Journal of texture studies*, Vol. 51, No. 2, pp. 232–241, 2020.
- [77] Têko Gouyo, Christian Mestres, Isabelle Maraval, Bénédicte Fontez, Céline Hofleitner, and Philippe Bohuon. Assessment of acoustic-mechanical measurements for texture of french fries: Comparison of deep-fat frying and air frying. *Food Research International*, Vol. 131, p. 108947, 2020.
- [78] Jacob Tizhe Liberty, Md Hafizur Rahman Bhuiyan, and Michael Ngadi. Assessing textural changes of breaded deep-fat fried chicken nuggets during post-frying holdings under infrared heat-lamp using acoustic-mechanical techniques. *International Journal of Food Science and Technology*, Vol. 59, No. 11, pp. 8596–8605, 2024.
- [79] Jacob Tizhe Liberty, Md Hafizur Rahman Bhuiyan, and Michael Ngadi. Unravelling the nexus between structure, texture, and acoustic traits of fried chicken nuggets.

*International Journal of Food Science and Technology*, Vol. 59, No. 11, pp. 8571–8582, 2024.

## 付録A 本研究に関する論文と発表実績

1. 新井 優作, ロペズ ギヨーム: 唐揚げの音響特性による揚げ終わり判定, マルチメディア, 分散, 協調とモバイル (DICOMO2023) シンポジウム論文集, pp.137-142(2023)

## 付録B 唐揚げの一般的なレシピまとめ

インターネット上に掲載されている 10 種類の唐揚げレシピを調査した結果を表 B.1 にまとめる。空欄となっている箇所はレシピに明記されていなかった情報である。

表 B.1: 唐揚げレシピの調理条件まとめ

	温度 [°C]	鶏もも肉の量 [g]	1 つ当たりの大きさ	調理時間 [分]	油の種類
レシピ 1	180	300	-	4	-
レシピ 2	160	250	-	4-5	サラダ油
レシピ 3	160-170	300-400	3-4 cm	4-6	-
レシピ 4	170	350	4-5 cm	6	-
レシピ 5	170	300	-	3-4	サラダ油
レシピ 6	170	100	小さめの一口大	3-4	-
レシピ 7	170-180	800	-	3	サラダ油
レシピ 8	170	250	3-4 cm 四方	4-6	サラダ油
レシピ 9	170	300	3-4 cm 程度	4-5	サラダ油
レシピ 10	160-170	300	一口大	4-5	-

補足として、レシピ 2 は 160°C で 3 分から 4 分調理した後、強火で 1 分程度揚げる。レシピ 3 は 3 分から 4 分調理した後、4 分から 5 分休める。その後、高温の油で 1 分から 2 分揚げる。レシピ 4 は 3 分フライ調理後、2 分裏返して調理、最後に強火で 1 分調理する。レシピ 5 の調理油は鍋の底から 3 cm 程度まで入れて調理する。レシピ 7 は 2 分程度揚げた後、余熱で 4 分程度おき、再度 200°C で約 1 分揚げる。レシピ 8 は、まず 2 分から 3 分程度調理する。鶏肉の下部に揚げ色がついたら裏返し、さらに 2 分 3 分揚げる。レシピ 9 の調理油は鍋の底から 2 cm 程度まで入れる。レシピ 10 では、4 分から 5 分揚げた後、180°C でカラッと揚げる。

表 B.1 より、調理時間は平均 4 分程度、調理油は基本的にサラダ油が使用されており、160°C から 180°C の温度で調理されることが一般的であることが判明した。

## 付 錄C MFCC特徴量を使用したその他の分類精度の評価

### C.1 アンダーサンプリングに Cluster Centroid を使用した際の分類精度一覧

表 C.1 はアンダーサンプリングに CC, オーバーサンプリングに BSMOTE を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle: Finish = 1,740: 1,441$  となった。

表 C.1: CC+BSMOTE を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.420 \pm 0.304$	$0.480 \pm 0.294$	$0.520 \pm 0.451$	$0.619 \pm 0.104$
ランダムフォレスト	$0.133 \pm 0.190$	$0.759 \pm 0.340$	$0.092 \pm 0.135$	$0.585 \pm 0.078$
GNB	$0.375 \pm 0.363$	$0.580 \pm 0.334$	$0.487 \pm 0.481$	$0.609 \pm 0.150$
kNN	$0.194 \pm 0.280$	$0.381 \pm 0.361$	$0.221 \pm 0.369$	$0.551 \pm 0.118$
$SVM_{linear}$	$0.482 \pm 0.266$	$0.616 \pm 0.211$	$0.618 \pm 0.425$	$0.608 \pm 0.110$
$SVM_{poly}$	$0.393 \pm 0.319$	$0.762 \pm 0.264$	$0.514 \pm 0.483$	$0.611 \pm 0.117$
$SVM_{rbf}$	$0.377 \pm 0.345$	$0.757 \pm 0.271$	$0.518 \pm 0.503$	$0.603 \pm 0.136$
$SVM_{sigmoid}$	$0.256 \pm 0.332$	$0.196 \pm 0.254$	$0.383 \pm 0.495$	$0.490 \pm 0.178$
XGB	$0.272 \pm 0.308$	$0.631 \pm 0.344$	$0.293 \pm 0.386$	$0.608 \pm 0.112$

表 C.2 はアンダーサンプリングに CC, オーバーサンプリングに  $SVMSMOTE_{linear}$  を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle: Finish = 1,740: 1,516$  となった。

表 C.2: CC+ $SVMSMOTE_{linear}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.420 \pm 0.304$	$0.480 \pm 0.294$	$0.520 \pm 0.451$	$0.619 \pm 0.104$
ランダムフォレスト	$0.137 \pm 0.260$	$0.745 \pm 0.362$	$0.119 \pm 0.264$	$0.600 \pm 0.105$
GNB	$0.375 \pm 0.363$	$0.580 \pm 0.334$	$0.487 \pm 0.481$	$0.609 \pm 0.150$
kNN	$0.194 \pm 0.280$	$0.381 \pm 0.361$	$0.221 \pm 0.369$	$0.551 \pm 0.118$
$SVM_{linear}$	$0.482 \pm 0.266$	$0.616 \pm 0.211$	$0.618 \pm 0.425$	$0.608 \pm 0.110$
$SVM_{poly}$	$0.393 \pm 0.319$	$0.762 \pm 0.264$	$0.514 \pm 0.483$	$0.611 \pm 0.117$
$SVM_{rbf}$	$0.377 \pm 0.345$	$0.757 \pm 0.271$	$0.518 \pm 0.503$	$0.603 \pm 0.136$
$SVM_{sigmoid}$	$0.256 \pm 0.332$	$0.196 \pm 0.254$	$0.383 \pm 0.495$	$0.490 \pm 0.178$
XGB	$0.272 \pm 0.308$	$0.631 \pm 0.344$	$0.293 \pm 0.386$	$0.608 \pm 0.112$

表C.3はアンダーサンプリングにCC, オーバーサンプリングに $SVMSMOTE_{poly}$ を適用した各分類器による分類精度である. クラスのサンプル数はMiddle:Finish = 1,740:1,575となった.

表 C.3: CC+ $SVMSMOTE_{poly}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.470 \pm 0.307$	$0.517 \pm 0.300$	$0.551 \pm 0.432$	$0.615 \pm 0.091$
ランダムフォレスト	$0.231 \pm 0.311$	$0.774 \pm 0.318$	$0.230 \pm 0.349$	$0.579 \pm 0.077$
GNB	$0.449 \pm 0.342$	$0.718 \pm 0.252$	$0.562 \pm 0.473$	$0.602 \pm 0.134$
kNN	$0.203 \pm 0.290$	$0.407 \pm 0.367$	$0.222 \pm 0.368$	$0.522 \pm 0.115$
$SVM_{linear}$	$0.511 \pm 0.278$	$0.553 \pm 0.265$	$0.588 \pm 0.402$	$0.618 \pm 0.106$
$SVM_{poly}$	$0.526 \pm 0.335$	$0.764 \pm 0.208$	$0.637 \pm 0.457$	$0.656 \pm 0.125$
$SVM_{rbf}$	$0.581 \pm 0.318$	$0.719 \pm 0.207$	$0.728 \pm 0.418$	$0.655 \pm 0.131$
$SVM_{sigmoid}$	$0.285 \pm 0.346$	$0.230 \pm 0.268$	$0.401 \pm 0.499$	$0.472 \pm 0.164$
XGB	$0.334 \pm 0.287$	$0.635 \pm 0.306$	$0.301 \pm 0.302$	$0.598 \pm 0.103$

表C.4はアンダーサンプリングにCC, オーバーサンプリングに $SVMSMOTE_{rbf}$ を適用した各分類器による分類精度である. クラスのサンプル数はMiddle:Finish = 1,740:1,575となった.

表 C.4: CC+ $SVMSMOTE_{rbf}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.455 \pm 0.305$	$0.511 \pm 0.296$	$0.537 \pm 0.438$	$0.602 \pm 0.091$
ランダムフォレスト	$0.231 \pm 0.323$	$0.802 \pm 0.310$	$0.206 \pm 0.321$	$0.591 \pm 0.119$
GNB	$0.447 \pm 0.343$	$0.717 \pm 0.255$	$0.560 \pm 0.475$	$0.601 \pm 0.134$
kNN	$0.208 \pm 0.295$	$0.408 \pm 0.369$	$0.227 \pm 0.372$	$0.524 \pm 0.119$
$SVM_{linear}$	$0.523 \pm 0.262$	$0.559 \pm 0.239$	$0.617 \pm 0.388$	$0.610 \pm 0.096$
$SVM_{poly}$	$0.527 \pm 0.331$	$0.763 \pm 0.212$	$0.637 \pm 0.453$	$0.653 \pm 0.124$
$SVM_{rbf}$	$0.573 \pm 0.317$	$0.717 \pm 0.208$	$0.721 \pm 0.423$	$0.650 \pm 0.128$
$SVM_{sigmoid}$	$0.283 \pm 0.344$	$0.229 \pm 0.266$	$0.397 \pm 0.494$	$0.470 \pm 0.164$
XGB	$0.364 \pm 0.283$	$0.654 \pm 0.299$	$0.330 \pm 0.301$	$0.606 \pm 0.099$

表C.5はアンダーサンプリングにCC, オーバーサンプリングに $SVMSMOTE_{sigmoid}$ を適用した各分類器による分類精度である. クラスのサンプル数はMiddle:Finish = 1,740:1,575となった.

表C.6はアンダーサンプリングにCC, オーバーサンプリングにADASYNを適用した各分類器による分類精度である. クラスのサンプル数はMiddle:Finish = 1,740:1,724となった.

表 C.5: CC+ $SVM_{sigmoid}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.453 \pm 0.299$	$0.518 \pm 0.287$	$0.533 \pm 0.437$	$0.596 \pm 0.091$
ランダムフォレスト	$0.208 \pm 0.289$	$0.772 \pm 0.312$	$0.189 \pm 0.289$	$0.565 \pm 0.096$
GNB	$0.444 \pm 0.343$	$0.717 \pm 0.255$	$0.556 \pm 0.477$	$0.600 \pm 0.135$
kNN	$0.210 \pm 0.295$	$0.412 \pm 0.372$	$0.228 \pm 0.372$	$0.524 \pm 0.119$
$SVM_{linear}$	$0.511 \pm 0.263$	$0.561 \pm 0.235$	$0.611 \pm 0.407$	$0.608 \pm 0.084$
$SVM_{poly}$	$0.528 \pm 0.329$	$0.764 \pm 0.212$	$0.636 \pm 0.450$	$0.654 \pm 0.124$
$SVM_{rbf}$	$0.574 \pm 0.317$	$0.716 \pm 0.208$	$0.723 \pm 0.421$	$0.651 \pm 0.126$
$SVM_{sigmoid}$	$0.282 \pm 0.343$	$0.229 \pm 0.266$	$0.395 \pm 0.492$	$0.469 \pm 0.164$
XGB	$0.368 \pm 0.268$	$0.734 \pm 0.222$	$0.356 \pm 0.330$	$0.592 \pm 0.089$

表 C.6: CC+ADASYN を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.453 \pm 0.299$	$0.518 \pm 0.287$	$0.533 \pm 0.437$	$0.596 \pm 0.091$
ランダムフォレスト	$0.178 \pm 0.304$	$0.821 \pm 0.315$	$0.166 \pm 0.313$	$0.582 \pm 0.094$
GNB	$0.444 \pm 0.343$	$0.717 \pm 0.255$	$0.556 \pm 0.477$	$0.600 \pm 0.135$
kNN	$0.210 \pm 0.295$	$0.412 \pm 0.372$	$0.228 \pm 0.372$	$0.524 \pm 0.119$
$SVM_{linear}$	$0.511 \pm 0.263$	$0.561 \pm 0.235$	$0.611 \pm 0.407$	$0.608 \pm 0.084$
$SVM_{poly}$	$0.528 \pm 0.329$	$0.764 \pm 0.212$	$0.636 \pm 0.450$	$0.654 \pm 0.124$
$SVM_{rbf}$	$0.574 \pm 0.317$	$0.716 \pm 0.208$	$0.723 \pm 0.421$	$0.651 \pm 0.126$
$SVM_{sigmoid}$	$0.282 \pm 0.343$	$0.229 \pm 0.266$	$0.395 \pm 0.492$	$0.469 \pm 0.164$
XGB	$0.368 \pm 0.268$	$0.734 \pm 0.222$	$0.356 \pm 0.330$	$0.592 \pm 0.089$

## C.2 アンダーサンプリングに Condensed Nearest Neighbour を使用した際の分類精度一覧

表 C.7 はアンダーサンプリングに CNN, オーバーサンプリングに BSMOTE を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle: Finish = 870: 870$  となった。

表 C.8 はアンダーサンプリングに CNN, オーバーサンプリングに  $SVM_{SMOTE}_{linear}$  を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle: Finish = 870: 870$  となった。

表 C.9 はアンダーサンプリングに CNN, オーバーサンプリングに  $SVM_{SMOTE}_{poly}$  を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle: Finish = 870: 870$  となった。

表 C.10 はアンダーサンプリングに CNN, オーバーサンプリングに  $SVM_{SMOTE}_{rbf}$  を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle: Finish = 870: 870$  となった。

表 C.11 はアンダーサンプリングに CNN, オーバーサンプリングに  $SVM_{SMOTE}_{sigmoid}$  を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle: Finish =$

表 C.7: CNN+BSMOTE を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.456 \pm 0.295$	$0.628 \pm 0.209$	$0.588 \pm 0.456$	$0.541 \pm 0.045$
ランダムフォレスト	$0.327 \pm 0.294$	$0.602 \pm 0.319$	$0.383 \pm 0.382$	$0.497 \pm 0.064$
GNB	$0.421 \pm 0.324$	$0.666 \pm 0.232$	$0.579 \pm 0.478$	$0.520 \pm 0.049$
kNN	$0.208 \pm 0.329$	$0.520 \pm 0.428$	$0.241 \pm 0.404$	$0.514 \pm 0.148$
$SVM_{linear}$	$0.459 \pm 0.278$	$0.577 \pm 0.195$	$0.586 \pm 0.445$	$0.535 \pm 0.039$
$SVM_{poly}$	$0.432 \pm 0.320$	$0.557 \pm 0.284$	$0.590 \pm 0.471$	$0.513 \pm 0.052$
$SVM_{rbf}$	$0.484 \pm 0.316$	$0.591 \pm 0.240$	$0.676 \pm 0.462$	$0.528 \pm 0.052$
$SVM_{sigmoid}$	$0.450 \pm 0.315$	$0.341 \pm 0.273$	$0.664 \pm 0.471$	$0.479 \pm 0.046$
XGB	$0.289 \pm 0.271$	$0.548 \pm 0.296$	$0.330 \pm 0.373$	$0.485 \pm 0.053$

表 C.8: CNN+SVMSMOTE<sub>linear</sub> を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.442 \pm 0.309$	$0.636 \pm 0.210$	$0.569 \pm 0.468$	$0.537 \pm 0.060$
ランダムフォレスト	$0.239 \pm 0.272$	$0.521 \pm 0.343$	$0.246 \pm 0.329$	$0.497 \pm 0.076$
GNB	$0.389 \pm 0.333$	$0.709 \pm 0.243$	$0.495 \pm 0.467$	$0.531 \pm 0.068$
kNN	$0.201 \pm 0.309$	$0.421 \pm 0.414$	$0.244 \pm 0.412$	$0.485 \pm 0.111$
$SVM_{linear}$	$0.456 \pm 0.279$	$0.656 \pm 0.205$	$0.569 \pm 0.454$	$0.541 \pm 0.053$
$SVM_{poly}$	$0.458 \pm 0.326$	$0.590 \pm 0.281$	$0.616 \pm 0.472$	$0.535 \pm 0.049$
$SVM_{rbf}$	$0.545 \pm 0.269$	$0.619 \pm 0.202$	$0.722 \pm 0.414$	$0.546 \pm 0.084$
$SVM_{sigmoid}$	$0.343 \pm 0.362$	$0.272 \pm 0.290$	$0.475 \pm 0.506$	$0.507 \pm 0.069$
XGB	$0.359 \pm 0.305$	$0.593 \pm 0.281$	$0.400 \pm 0.399$	$0.538 \pm 0.071$

870:870 となった。

### C.3 アンダーサンプリングに NearMiss-1 を使用した際の分類精度一覧

表 C.12 はアンダーサンプリングに NM1, オーバーサンプリングに BSMOTE を適用した各分類器による分類精度である。クラスのサンプル数は Middle:Finish = 1,740:1,740 となった。

表 C.13 はアンダーサンプリングに NM1, オーバーサンプリングに SVMSMOTE<sub>linear</sub> を適用した各分類器による分類精度である。クラスのサンプル数は Middle:Finish = 1,740:1,740 となった。

表 C.14 はアンダーサンプリングに NM1, オーバーサンプリングに SVMSMOTE<sub>poly</sub> を適用した各分類器による分類精度である。クラスのサンプル数は Middle:Finish = 1,740:1,740 となった。

表 C.15 はアンダーサンプリングに NM1, オーバーサンプリングに SVMSMOTE<sub>rbf</sub> を適用した各分類器による分類精度である。クラスのサンプル数は Middle:Finish = 1,740:1,740 となった。

表 C.16 はアンダーサンプリングに NM1, オーバーサンプリングに SVMSMOTE<sub>sigmoid</sub>

表 C.9: CNN+ $SVM_{SMOTE_{poly}}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	0.448 ± 0.318	0.635 ± 0.160	0.576 ± 0.474	0.556 ± 0.070
ランダムフォレスト	0.273 ± 0.259	0.545 ± 0.349	0.266 ± 0.311	0.520 ± 0.068
GNB	0.434 ± 0.331	0.658 ± 0.249	0.578 ± 0.478	0.540 ± 0.076
kNN	0.199 ± 0.322	0.508 ± 0.427	0.237 ± 0.406	0.499 ± 0.151
$SVM_{linear}$	0.443 ± 0.295	0.636 ± 0.157	0.563 ± 0.469	0.548 ± 0.049
$SVM_{poly}$	0.517 ± 0.293	0.637 ± 0.200	0.681 ± 0.426	0.552 ± 0.079
$SVM_{rbf}$	0.557 ± 0.289	0.640 ± 0.197	0.766 ± 0.407	0.561 ± 0.071
$SVM_{sigmoid}$	0.313 ± 0.336	0.260 ± 0.278	0.415 ± 0.464	0.493 ± 0.067
XGB	0.322 ± 0.277	0.475 ± 0.308	0.334 ± 0.317	0.507 ± 0.072

表 C.10: CNN+ $SVM_{SMOTE_{rbf}}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	0.448 ± 0.318	0.635 ± 0.160	0.576 ± 0.474	0.556 ± 0.070
ランダムフォレスト	0.288 ± 0.270	0.589 ± 0.328	0.296 ± 0.309	0.500 ± 0.076
GNB	0.434 ± 0.331	0.658 ± 0.249	0.578 ± 0.478	0.540 ± 0.076
kNN	0.199 ± 0.322	0.508 ± 0.427	0.237 ± 0.406	0.499 ± 0.151
$SVM_{linear}$	0.443 ± 0.295	0.636 ± 0.157	0.563 ± 0.469	0.548 ± 0.049
$SVM_{poly}$	0.517 ± 0.293	0.637 ± 0.200	0.681 ± 0.426	0.552 ± 0.079
$SVM_{rbf}$	0.557 ± 0.289	0.640 ± 0.197	0.766 ± 0.407	0.561 ± 0.071
$SVM_{sigmoid}$	0.313 ± 0.336	0.260 ± 0.278	0.415 ± 0.464	0.493 ± 0.067
XGB	0.322 ± 0.277	0.475 ± 0.308	0.334 ± 0.317	0.507 ± 0.072

を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle:Finish = 1,740:1,740$  となった。

表 C.17 はアンダーサンプリングに NM1, オーバーサンプリングに ADASYN を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle:Finish = 1,740:1,726$  となった。

#### C.4 アンダーサンプリングに NearMiss-2 を使用した際の分類精度一覧

表 C.18 はアンダーサンプリングに NM2, オーバーサンプリングに BSMOTE を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle:Finish = 1,740:1,740$  となった。

表 C.19 はアンダーサンプリングに NM2, オーバーサンプリングに  $SVM_{SMOTE_{linear}}$  を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle:Finish = 1,740:1,740$  となった。

表 C.20 はアンダーサンプリングに NM2, オーバーサンプリングに  $SVM_{SMOTE_{poly}}$  を適用した各分類器による分類精度である。クラスのサンプル数は  $Middle:Finish = 1,740:1,740$  となった。

表 C.11: CNN+ $SVMSMOTE_{sigmoid}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	0.448 ± 0.318	0.635 ± 0.160	0.576 ± 0.474	0.556 ± 0.070
ランダムフォレスト	0.300 ± 0.277	0.500 ± 0.340	0.308 ± 0.316	0.509 ± 0.075
GNB	0.434 ± 0.331	0.658 ± 0.249	0.578 ± 0.478	0.540 ± 0.076
kNN	0.199 ± 0.322	0.508 ± 0.427	0.237 ± 0.406	0.499 ± 0.151
$SVM_{linear}$	0.443 ± 0.295	0.636 ± 0.157	0.563 ± 0.469	0.548 ± 0.049
$SVM_{poly}$	0.517 ± 0.293	0.637 ± 0.200	0.681 ± 0.426	0.552 ± 0.079
$SVM_{rbf}$	0.557 ± 0.289	0.640 ± 0.197	0.766 ± 0.407	0.561 ± 0.071
$SVM_{sigmoid}$	0.313 ± 0.336	0.260 ± 0.278	0.415 ± 0.464	0.493 ± 0.067
XGB	0.322 ± 0.277	0.475 ± 0.308	0.334 ± 0.317	0.507 ± 0.072

表 C.12: NM1+BSMOTE を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	0.401 ± 0.299	0.642 ± 0.201	0.530 ± 0.468	0.513 ± 0.015
ランダムフォレスト	0.163 ± 0.218	0.554 ± 0.363	0.164 ± 0.290	0.484 ± 0.052
GNB	0.410 ± 0.355	0.722 ± 0.242	0.572 ± 0.499	0.535 ± 0.063
kNN	0.272 ± 0.314	0.690 ± 0.345	0.343 ± 0.430	0.503 ± 0.089
$SVM_{linear}$	0.447 ± 0.283	0.658 ± 0.199	0.574 ± 0.455	0.532 ± 0.042
$SVM_{poly}$	0.464 ± 0.320	0.576 ± 0.284	0.631 ± 0.461	0.530 ± 0.067
$SVM_{rbf}$	0.474 ± 0.297	0.574 ± 0.252	0.671 ± 0.455	0.487 ± 0.088
$SVM_{sigmoid}$	0.590 ± 0.212	0.558 ± 0.157	0.847 ± 0.329	0.513 ± 0.042
XGB	0.277 ± 0.225	0.553 ± 0.309	0.262 ± 0.295	0.491 ± 0.075

表 C.21 はアンダーサンプリングに NM2, オーバーサンプリングに  $SVMSMOTE_{rbf}$  を適用した各分類器による分類精度である. クラスのサンプル数は  $Middle: Finish = 1,740: 1,740$  となった.

表 C.22 はアンダーサンプリングに NM2, オーバーサンプリングに  $SVMSMOTE_{sigmoid}$  を適用した各分類器による分類精度である. クラスのサンプル数は  $Middle: Finish = 1,740: 1,740$  となった.

表 C.23 はアンダーサンプリングに NM2, オーバーサンプリングに ADASYN を適用した各分類器による分類精度である. クラスのサンプル数は  $Middle: Finish = 1,740: 1,725$  となった.

## C.5 アンダーサンプリングに One-Sided Selection を使用した際の分類精度一覧

表 C.24 はアンダーサンプリングに OSS, オーバーサンプリングに BSMOTE を適用した各分類器による分類精度である. クラスのサンプル数は  $Middle: Finish = 6,470: 6,470$  となった.

表 C.25 はアンダーサンプリングに OSS, オーバーサンプリングに  $SVMSMOTE_{linear}$

表 C.13: NM1+ $SVMSMOTE_{linear}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	0.401 ± 0.291	0.616 ± 0.161	0.518 ± 0.461	0.517 ± 0.029
ランダムフォレスト	0.282 ± 0.235	0.642 ± 0.259	0.281 ± 0.294	0.491 ± 0.044
GNB	0.435 ± 0.329	0.568 ± 0.283	0.610 ± 0.494	0.523 ± 0.059
kNN	0.257 ± 0.303	0.688 ± 0.343	0.320 ± 0.415	0.495 ± 0.080
$SVM_{linear}$	0.425 ± 0.277	0.597 ± 0.157	0.539 ± 0.453	0.524 ± 0.043
$SVM_{poly}$	0.450 ± 0.329	0.678 ± 0.235	0.598 ± 0.470	0.540 ± 0.095
$SVM_{rbf}$	0.453 ± 0.314	0.555 ± 0.283	0.644 ± 0.476	0.497 ± 0.054
$SVM_{sigmoid}$	0.277 ± 0.335	0.666 ± 0.334	0.405 ± 0.509	0.470 ± 0.098
XGB	0.329 ± 0.240	0.536 ± 0.278	0.355 ± 0.332	0.467 ± 0.070

表 C.14: NM1+ $SVMSMOTE_{poly}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	0.426 ± 0.283	0.642 ± 0.172	0.552 ± 0.458	0.522 ± 0.024
ランダムフォレスト	0.274 ± 0.272	0.722 ± 0.247	0.301 ± 0.362	0.508 ± 0.024
GNB	0.419 ± 0.343	0.595 ± 0.291	0.593 ± 0.501	0.527 ± 0.057
kNN	0.266 ± 0.312	0.585 ± 0.390	0.337 ± 0.428	0.497 ± 0.099
$SVM_{linear}$	0.448 ± 0.265	0.640 ± 0.167	0.564 ± 0.443	0.530 ± 0.031
$SVM_{poly}$	0.454 ± 0.318	0.566 ± 0.283	0.627 ± 0.467	0.522 ± 0.052
$SVM_{rbf}$	0.473 ± 0.293	0.579 ± 0.240	0.668 ± 0.454	0.492 ± 0.055
$SVM_{sigmoid}$	0.281 ± 0.334	0.672 ± 0.334	0.407 ± 0.508	0.470 ± 0.103
XGB	0.346 ± 0.285	0.586 ± 0.263	0.390 ± 0.407	0.499 ± 0.130

を適用した各分類器による分類精度である. クラスのサンプル数は  $Middle: Finish = 6,470: 5,889$  となった.

表 C.26 はアンダーサンプリングに OSS, オーバーサンプリングに  $SVMSMOTE_{poly}$  を適用した各分類器による分類精度である. クラスのサンプル数は  $Middle: Finish = 6,470: 6,470$  となった.

表 C.27 はアンダーサンプリングに OSS, オーバーサンプリングに  $SVMSMOTE_{rbf}$  を適用した各分類器による分類精度である. クラスのサンプル数は  $Middle: Finish = 6,470: 6,470$  となった.

表 C.28 はアンダーサンプリングに OSS, オーバーサンプリングに  $SVMSMOTE_{sigmoid}$  を適用した各分類器による分類精度である. クラスのサンプル数は  $Middle: Finish = 6,470: 6,470$  となった.

表 C.29 はアンダーサンプリングに NM2, オーバーサンプリングに ADASYN を適用した各分類器による分類精度である. クラスのサンプル数は  $Middle: Finish = 6,470: 6,499$  となった.

表 C.15: NM1+ $SVM_{SMOTE_{rbf}}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.425 \pm 0.279$	$0.644 \pm 0.176$	$0.552 \pm 0.460$	$0.521 \pm 0.021$
ランダムフォレスト	$0.364 \pm 0.241$	$0.651 \pm 0.248$	$0.398 \pm 0.337$	$0.500 \pm 0.049$
GNB	$0.421 \pm 0.340$	$0.599 \pm 0.294$	$0.592 \pm 0.498$	$0.528 \pm 0.058$
kNN	$0.267 \pm 0.313$	$0.586 \pm 0.390$	$0.338 \pm 0.430$	$0.497 \pm 0.101$
$SVM_{linear}$	$0.450 \pm 0.267$	$0.647 \pm 0.172$	$0.575 \pm 0.456$	$0.531 \pm 0.032$
$SVM_{poly}$	$0.446 \pm 0.323$	$0.558 \pm 0.284$	$0.630 \pm 0.485$	$0.517 \pm 0.049$
$SVM_{rbf}$	$0.450 \pm 0.309$	$0.556 \pm 0.275$	$0.645 \pm 0.475$	$0.491 \pm 0.035$
$SVM_{sigmoid}$	$0.281 \pm 0.333$	$0.672 \pm 0.334$	$0.406 \pm 0.507$	$0.469 \pm 0.103$
XGB	$0.361 \pm 0.295$	$0.555 \pm 0.309$	$0.393 \pm 0.393$	$0.508 \pm 0.142$

表 C.16: NM1+ $SVM_{SMOTE_{sigmoid}}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.424 \pm 0.281$	$0.643 \pm 0.175$	$0.551 \pm 0.462$	$0.520 \pm 0.020$
ランダムフォレスト	$0.290 \pm 0.240$	$0.619 \pm 0.237$	$0.300 \pm 0.338$	$0.496 \pm 0.073$
GNB	$0.423 \pm 0.338$	$0.601 \pm 0.296$	$0.594 \pm 0.497$	$0.528 \pm 0.058$
kNN	$0.267 \pm 0.314$	$0.586 \pm 0.390$	$0.339 \pm 0.431$	$0.498 \pm 0.101$
$SVM_{linear}$	$0.443 \pm 0.267$	$0.644 \pm 0.170$	$0.567 \pm 0.457$	$0.528 \pm 0.028$
$SVM_{poly}$	$0.442 \pm 0.324$	$0.556 \pm 0.284$	$0.626 \pm 0.489$	$0.515 \pm 0.046$
$SVM_{rbf}$	$0.441 \pm 0.312$	$0.552 \pm 0.274$	$0.634 \pm 0.482$	$0.490 \pm 0.027$
$SVM_{sigmoid}$	$0.281 \pm 0.333$	$0.672 \pm 0.334$	$0.406 \pm 0.507$	$0.469 \pm 0.103$
XGB	$0.392 \pm 0.265$	$0.612 \pm 0.237$	$0.410 \pm 0.359$	$0.531 \pm 0.111$

表 C.17: NM1+ADASYN を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.401 \pm 0.294$	$0.653 \pm 0.204$	$0.527 \pm 0.470$	$0.517 \pm 0.019$
ランダムフォレスト	$0.223 \pm 0.217$	$0.589 \pm 0.274$	$0.206 \pm 0.280$	$0.488 \pm 0.085$
GNB	$0.419 \pm 0.345$	$0.603 \pm 0.289$	$0.572 \pm 0.491$	$0.539 \pm 0.066$
kNN	$0.266 \pm 0.314$	$0.585 \pm 0.386$	$0.339 \pm 0.431$	$0.498 \pm 0.091$
$SVM_{linear}$	$0.437 \pm 0.289$	$0.654 \pm 0.199$	$0.563 \pm 0.464$	$0.532 \pm 0.046$
$SVM_{poly}$	$0.448 \pm 0.321$	$0.559 \pm 0.283$	$0.629 \pm 0.485$	$0.521 \pm 0.039$
$SVM_{rbf}$	$0.454 \pm 0.311$	$0.555 \pm 0.283$	$0.646 \pm 0.474$	$0.500 \pm 0.038$
$SVM_{sigmoid}$	$0.578 \pm 0.218$	$0.549 \pm 0.161$	$0.833 \pm 0.344$	$0.503 \pm 0.044$
XGB	$0.268 \pm 0.256$	$0.518 \pm 0.327$	$0.297 \pm 0.343$	$0.466 \pm 0.068$

表 C.18: NM2+BSMOTE を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.385 \pm 0.287$	$0.607 \pm 0.217$	$0.494 \pm 0.456$	$0.504 \pm 0.027$
ランダムフォレスト	$0.138 \pm 0.217$	$0.576 \pm 0.384$	$0.124 \pm 0.243$	$0.512 \pm 0.039$
GNB	$0.439 \pm 0.377$	$0.676 \pm 0.293$	$0.577 \pm 0.493$	$0.562 \pm 0.142$
kNN	$0.242 \pm 0.281$	$0.473 \pm 0.359$	$0.272 \pm 0.369$	$0.478 \pm 0.098$
$SVM_{linear}$	$0.445 \pm 0.271$	$0.644 \pm 0.201$	$0.551 \pm 0.426$	$0.528 \pm 0.051$
$SVM_{poly}$	$0.499 \pm 0.269$	$0.616 \pm 0.209$	$0.664 \pm 0.395$	$0.512 \pm 0.059$
$SVM_{rbf}$	$0.524 \pm 0.280$	$0.608 \pm 0.207$	$0.752 \pm 0.423$	$0.513 \pm 0.017$
$SVM_{sigmoid}$	$0.589 \pm 0.215$	$0.557 \pm 0.158$	$0.844 \pm 0.331$	$0.514 \pm 0.045$
XGB	$0.197 \pm 0.249$	$0.510 \pm 0.382$	$0.217 \pm 0.342$	$0.488 \pm 0.039$

表 C.19: NM2+ $SVMSMOTE_{linear}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.384 \pm 0.279$	$0.606 \pm 0.215$	$0.486 \pm 0.452$	$0.505 \pm 0.019$
ランダムフォレスト	$0.168 \pm 0.205$	$0.695 \pm 0.274$	$0.149 \pm 0.253$	$0.505 \pm 0.031$
GNB	$0.456 \pm 0.357$	$0.581 \pm 0.310$	$0.609 \pm 0.487$	$0.549 \pm 0.147$
kNN	$0.216 \pm 0.257$	$0.553 \pm 0.354$	$0.236 \pm 0.337$	$0.459 \pm 0.093$
$SVM_{linear}$	$0.426 \pm 0.259$	$0.637 \pm 0.203$	$0.520 \pm 0.425$	$0.518 \pm 0.029$
$SVM_{poly}$	$0.442 \pm 0.285$	$0.575 \pm 0.245$	$0.588 \pm 0.441$	$0.483 \pm 0.099$
$SVM_{rbf}$	$0.520 \pm 0.280$	$0.605 \pm 0.208$	$0.746 \pm 0.427$	$0.511 \pm 0.021$
$SVM_{sigmoid}$	$0.583 \pm 0.217$	$0.551 \pm 0.160$	$0.837 \pm 0.339$	$0.507 \pm 0.042$
XGB	$0.321 \pm 0.283$	$0.544 \pm 0.318$	$0.371 \pm 0.408$	$0.491 \pm 0.087$

表 C.20: NM2+ $SVMSMOTE_{poly}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.395 \pm 0.279$	$0.617 \pm 0.215$	$0.503 \pm 0.450$	$0.507 \pm 0.026$
ランダムフォレスト	$0.213 \pm 0.220$	$0.620 \pm 0.327$	$0.184 \pm 0.243$	$0.510 \pm 0.060$
GNB	$0.459 \pm 0.351$	$0.694 \pm 0.251$	$0.598 \pm 0.471$	$0.552 \pm 0.145$
kNN	$0.256 \pm 0.287$	$0.472 \pm 0.360$	$0.296 \pm 0.380$	$0.468 \pm 0.103$
$SVM_{linear}$	$0.433 \pm 0.260$	$0.638 \pm 0.200$	$0.534 \pm 0.426$	$0.519 \pm 0.029$
$SVM_{poly}$	$0.466 \pm 0.290$	$0.591 \pm 0.243$	$0.617 \pm 0.433$	$0.498 \pm 0.107$
$SVM_{rbf}$	$0.523 \pm 0.280$	$0.607 \pm 0.207$	$0.751 \pm 0.423$	$0.511 \pm 0.013$
$SVM_{sigmoid}$	$0.513 \pm 0.282$	$0.600 \pm 0.214$	$0.731 \pm 0.425$	$0.506 \pm 0.046$
XGB	$0.371 \pm 0.301$	$0.468 \pm 0.303$	$0.420 \pm 0.418$	$0.536 \pm 0.105$

表 C.21: NM2+ $SVM_{SMOTE_{rbf}}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.399 \pm 0.279$	$0.623 \pm 0.213$	$0.507 \pm 0.450$	$0.510 \pm 0.018$
ランダムフォレスト	$0.137 \pm 0.211$	$0.615 \pm 0.379$	$0.137 \pm 0.278$	$0.496 \pm 0.033$
GNB	$0.464 \pm 0.351$	$0.697 \pm 0.248$	$0.604 \pm 0.470$	$0.554 \pm 0.147$
kNN	$0.258 \pm 0.288$	$0.473 \pm 0.359$	$0.298 \pm 0.382$	$0.469 \pm 0.105$
$SVM_{linear}$	$0.436 \pm 0.259$	$0.646 \pm 0.199$	$0.535 \pm 0.426$	$0.523 \pm 0.027$
$SVM_{poly}$	$0.443 \pm 0.307$	$0.574 \pm 0.266$	$0.593 \pm 0.458$	$0.498 \pm 0.096$
$SVM_{rbf}$	$0.507 \pm 0.282$	$0.597 \pm 0.214$	$0.728 \pm 0.434$	$0.501 \pm 0.024$
$SVM_{sigmoid}$	$0.578 \pm 0.220$	$0.549 \pm 0.163$	$0.828 \pm 0.344$	$0.505 \pm 0.047$
XGB	$0.328 \pm 0.266$	$0.550 \pm 0.317$	$0.353 \pm 0.375$	$0.508 \pm 0.069$

表 C.22: NM2+ $SVM_{SMOTE_{sigmoid}}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.397 \pm 0.280$	$0.623 \pm 0.211$	$0.504 \pm 0.450$	$0.510 \pm 0.018$
ランダムフォレスト	$0.138 \pm 0.175$	$0.581 \pm 0.329$	$0.110 \pm 0.165$	$0.492 \pm 0.028$
GNB	$0.468 \pm 0.350$	$0.697 \pm 0.248$	$0.610 \pm 0.465$	$0.554 \pm 0.148$
kNN	$0.259 \pm 0.290$	$0.473 \pm 0.360$	$0.300 \pm 0.385$	$0.470 \pm 0.106$
$SVM_{linear}$	$0.432 \pm 0.259$	$0.644 \pm 0.198$	$0.531 \pm 0.426$	$0.522 \pm 0.024$
$SVM_{poly}$	$0.430 \pm 0.318$	$0.561 \pm 0.284$	$0.582 \pm 0.471$	$0.501 \pm 0.083$
$SVM_{rbf}$	$0.492 \pm 0.282$	$0.588 \pm 0.220$	$0.706 \pm 0.437$	$0.491 \pm 0.039$
$SVM_{sigmoid}$	$0.578 \pm 0.220$	$0.549 \pm 0.163$	$0.828 \pm 0.344$	$0.505 \pm 0.047$
XGB	$0.332 \pm 0.255$	$0.469 \pm 0.308$	$0.359 \pm 0.371$	$0.508 \pm 0.052$

表 C.23: NM2+ASASYN を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.372 \pm 0.284$	$0.602 \pm 0.221$	$0.475 \pm 0.459$	$0.501 \pm 0.036$
ランダムフォレスト	$0.122 \pm 0.214$	$0.683 \pm 0.366$	$0.117 \pm 0.251$	$0.504 \pm 0.040$
GNB	$0.451 \pm 0.360$	$0.592 \pm 0.318$	$0.580 \pm 0.473$	$0.556 \pm 0.149$
kNN	$0.249 \pm 0.282$	$0.472 \pm 0.352$	$0.284 \pm 0.377$	$0.470 \pm 0.111$
$SVM_{linear}$	$0.409 \pm 0.274$	$0.632 \pm 0.208$	$0.507 \pm 0.442$	$0.516 \pm 0.041$
$SVM_{poly}$	$0.432 \pm 0.313$	$0.561 \pm 0.284$	$0.582 \pm 0.464$	$0.500 \pm 0.075$
$SVM_{rbf}$	$0.517 \pm 0.283$	$0.605 \pm 0.209$	$0.739 \pm 0.432$	$0.514 \pm 0.025$
$SVM_{sigmoid}$	$0.579 \pm 0.222$	$0.550 \pm 0.164$	$0.831 \pm 0.347$	$0.508 \pm 0.056$
XGB	$0.180 \pm 0.251$	$0.660 \pm 0.300$	$0.198 \pm 0.333$	$0.492 \pm 0.032$

表 C.24: OSS+BSMOTE を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.546 \pm 0.284$	$0.678 \pm 0.232$	$0.600 \pm 0.388$	$0.625 \pm 0.103$
ランダムフォレスト	$0.041 \pm 0.067$	$0.762 \pm 0.362$	$0.025 \pm 0.041$	$0.496 \pm 0.024$
GNB	$0.392 \pm 0.403$	$0.674 \pm 0.342$	$0.462 \pm 0.487$	$0.601 \pm 0.143$
kNN	$0.183 \pm 0.312$	$0.511 \pm 0.438$	$0.205 \pm 0.382$	$0.495 \pm 0.147$
$SVM_{linear}$	$0.551 \pm 0.299$	$0.560 \pm 0.294$	$0.618 \pm 0.393$	$0.625 \pm 0.119$
$SVM_{poly}$	$0.397 \pm 0.358$	$0.675 \pm 0.205$	$0.480 \pm 0.476$	$0.575 \pm 0.117$
$SVM_{rbf}$	$0.504 \pm 0.331$	$0.695 \pm 0.182$	$0.627 \pm 0.462$	$0.603 \pm 0.109$
$SVM_{sigmoid}$	$0.430 \pm 0.385$	$0.672 \pm 0.309$	$0.538 \pm 0.499$	$0.593 \pm 0.141$
XGB	$0.151 \pm 0.242$	$0.699 \pm 0.366$	$0.138 \pm 0.255$	$0.506 \pm 0.081$

表 C.25: OSS+SVMSMOTE<sub>linear</sub> を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.525 \pm 0.222$	$0.593 \pm 0.256$	$0.581 \pm 0.360$	$0.600 \pm 0.089$
ランダムフォレスト	$0.051 \pm 0.089$	$0.722 \pm 0.394$	$0.033 \pm 0.057$	$0.502 \pm 0.071$
GNB	$0.324 \pm 0.377$	$0.606 \pm 0.391$	$0.409 \pm 0.489$	$0.558 \pm 0.153$
kNN	$0.174 \pm 0.299$	$0.493 \pm 0.418$	$0.190 \pm 0.359$	$0.504 \pm 0.124$
$SVM_{linear}$	$0.575 \pm 0.234$	$0.601 \pm 0.259$	$0.646 \pm 0.333$	$0.624 \pm 0.111$
$SVM_{poly}$	$0.384 \pm 0.328$	$0.656 \pm 0.225$	$0.480 \pm 0.467$	$0.555 \pm 0.113$
$SVM_{rbf}$	$0.503 \pm 0.298$	$0.670 \pm 0.205$	$0.639 \pm 0.431$	$0.582 \pm 0.101$
$SVM_{sigmoid}$	$0.394 \pm 0.378$	$0.641 \pm 0.330$	$0.520 \pm 0.509$	$0.569 \pm 0.158$
XGB	$0.176 \pm 0.191$	$0.568 \pm 0.400$	$0.141 \pm 0.153$	$0.504 \pm 0.055$

表 C.26: OSS+SVMSMOTE<sub>poly</sub> を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.573 \pm 0.266$	$0.622 \pm 0.252$	$0.629 \pm 0.375$	$0.641 \pm 0.101$
ランダムフォレスト	$0.073 \pm 0.123$	$0.627 \pm 0.432$	$0.052 \pm 0.086$	$0.480 \pm 0.054$
GNB	$0.383 \pm 0.399$	$0.603 \pm 0.338$	$0.469 \pm 0.494$	$0.584 \pm 0.139$
kNN	$0.207 \pm 0.312$	$0.529 \pm 0.426$	$0.232 \pm 0.386$	$0.493 \pm 0.142$
$SVM_{linear}$	$0.586 \pm 0.260$	$0.606 \pm 0.255$	$0.647 \pm 0.366$	$0.641 \pm 0.106$
$SVM_{poly}$	$0.492 \pm 0.333$	$0.733 \pm 0.177$	$0.586 \pm 0.449$	$0.610 \pm 0.098$
$SVM_{rbf}$	$0.553 \pm 0.311$	$0.706 \pm 0.181$	$0.692 \pm 0.431$	$0.615 \pm 0.093$
$SVM_{sigmoid}$	$0.424 \pm 0.384$	$0.667 \pm 0.310$	$0.533 \pm 0.502$	$0.589 \pm 0.140$
XGB	$0.169 \pm 0.243$	$0.695 \pm 0.360$	$0.164 \pm 0.267$	$0.499 \pm 0.074$

表 C.27: OSS+ $SVMSMOTE_{rbf}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.569 \pm 0.266$	$0.623 \pm 0.252$	$0.626 \pm 0.379$	$0.639 \pm 0.100$
ランダムフォレスト	$0.083 \pm 0.135$	$0.740 \pm 0.369$	$0.061 \pm 0.098$	$0.488 \pm 0.042$
GNB	$0.387 \pm 0.400$	$0.609 \pm 0.329$	$0.474 \pm 0.496$	$0.585 \pm 0.143$
kNN	$0.208 \pm 0.312$	$0.528 \pm 0.426$	$0.234 \pm 0.388$	$0.493 \pm 0.142$
$SVM_{linear}$	$0.578 \pm 0.263$	$0.604 \pm 0.255$	$0.640 \pm 0.372$	$0.638 \pm 0.105$
$SVM_{poly}$	$0.493 \pm 0.333$	$0.736 \pm 0.182$	$0.588 \pm 0.450$	$0.609 \pm 0.099$
$SVM_{rbf}$	$0.553 \pm 0.312$	$0.706 \pm 0.181$	$0.694 \pm 0.432$	$0.615 \pm 0.093$
$SVM_{sigmoid}$	$0.425 \pm 0.383$	$0.668 \pm 0.310$	$0.534 \pm 0.501$	$0.590 \pm 0.140$
XGB	$0.234 \pm 0.279$	$0.579 \pm 0.374$	$0.232 \pm 0.308$	$0.519 \pm 0.080$

表 C.28: OSS+ $SVMSMOTE_{sigmoid}$  を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.568 \pm 0.265$	$0.623 \pm 0.252$	$0.624 \pm 0.379$	$0.639 \pm 0.099$
ランダムフォレスト	$0.067 \pm 0.113$	$0.723 \pm 0.389$	$0.047 \pm 0.078$	$0.480 \pm 0.055$
GNB	$0.389 \pm 0.400$	$0.619 \pm 0.325$	$0.475 \pm 0.496$	$0.586 \pm 0.145$
kNN	$0.209 \pm 0.313$	$0.528 \pm 0.426$	$0.234 \pm 0.389$	$0.493 \pm 0.142$
$SVM_{linear}$	$0.576 \pm 0.263$	$0.604 \pm 0.255$	$0.637 \pm 0.372$	$0.637 \pm 0.105$
$SVM_{poly}$	$0.492 \pm 0.334$	$0.736 \pm 0.182$	$0.587 \pm 0.450$	$0.609 \pm 0.100$
$SVM_{rbf}$	$0.553 \pm 0.312$	$0.707 \pm 0.181$	$0.693 \pm 0.433$	$0.616 \pm 0.094$
$SVM_{sigmoid}$	$0.425 \pm 0.383$	$0.667 \pm 0.310$	$0.534 \pm 0.501$	$0.590 \pm 0.140$
XGB	$0.217 \pm 0.252$	$0.564 \pm 0.373$	$0.205 \pm 0.263$	$0.507 \pm 0.061$

表 C.29: OSS+ADASYN を適用した MFCC 特徴量の分類精度

分類器	F1 スコア	適合率	再現率	正解率
ロジスティック回帰	$0.547 \pm 0.249$	$0.611 \pm 0.252$	$0.593 \pm 0.379$	$0.622 \pm 0.088$
ランダムフォレスト	$0.058 \pm 0.101$	$0.733 \pm 0.380$	$0.038 \pm 0.065$	$0.490 \pm 0.031$
GNB	$0.403 \pm 0.403$	$0.680 \pm 0.324$	$0.481 \pm 0.493$	$0.597 \pm 0.158$
kNN	$0.192 \pm 0.307$	$0.498 \pm 0.401$	$0.216 \pm 0.377$	$0.489 \pm 0.135$
$SVM_{linear}$	$0.549 \pm 0.251$	$0.593 \pm 0.256$	$0.603 \pm 0.378$	$0.616 \pm 0.094$
$SVM_{poly}$	$0.451 \pm 0.338$	$0.711 \pm 0.180$	$0.545 \pm 0.476$	$0.592 \pm 0.105$
$SVM_{rbf}$	$0.544 \pm 0.307$	$0.699 \pm 0.178$	$0.677 \pm 0.434$	$0.610 \pm 0.093$
$SVM_{sigmoid}$	$0.422 \pm 0.383$	$0.666 \pm 0.310$	$0.532 \pm 0.502$	$0.587 \pm 0.141$
XGB	$0.108 \pm 0.165$	$0.477 \pm 0.433$	$0.085 \pm 0.140$	$0.477 \pm 0.078$

## 質疑応答

浦垣 啓志郎 情報テクノロジー学科 助手

Q	近年の研究では、CNNなどを利用した方が結果が出ることがわかっているにも関わらず、SVMなどの古典的な分類器を利用しているのはなぜですか。
A	ご質問ありがとうございます。本研究で取得したデータセットの問題があったためです。10種類の調理音データを取得しましたが、時系列データとしては、10種類のデータではデータ量として少ないため、CNNやRNNなどは検討できませんでした。実際に、LSTMを考慮いたしましたが、モデルの学習途中で過学習となっていることがすぐにわかり、断念しております。

浦垣 啓志郎 情報テクノロジー学科 助手

Q	解いている問題が非常に面白いです。ヒューリスティックアルゴリズムだと唐揚げ粉の材質とかにも敏感なので、対策に関する考察があると良いと思いました。
A	貴重なご意見ありがとうございます。本研究で利用した調理データには、パン粉で調理したものと唐揚げ粉を利用したものの2種類ございました。両者の特徴は多少異なりましたが、どちらも変化点検出を利用して揚げ終わりを捉えられておりました。今後、他の材料を使用した際の特徴の違いも考慮したシステムを構築していくたいと考えております。

大原 剛三 情報テクノロジー学科 教授

Q	変化点検出において、実際の検出されたタイミングと真値ラベルとの誤差は、何秒程度ありますか。
A	ご質問ありがとうございます。10種類の内、変化点を検出できたのは9種類であり、それらの真値ラベルと検出された変化点の差は、平均1.54秒でした。150秒以降に初めて検出された点という制限を設けることで、適合率も90%となっているため、安全な調理支援が可能であると考えています。

Dürst Martin 情報テクノロジー学科 教授

Q	Can your system be used to cook other things? (I wouldn't want to eat fried chicken every day.) How much time would the tuning for a different dish take, and what kinds of dishes would be suited?
A	ご質問ありがとうございます。現状のシステムでは、他の料理への適用は困難であると考えております。唐揚げのフライ調理の音響特徴に対して有効な変化点検出となっているため、他の料理に適用するためには、本研究のフロート同様に適切な特徴を調査し、精度検証を実施する必要があります。しかし、揚げ物については解析の結果、類似した音響特徴であると判明すれば、簡単に適用可能であると考えられます。またシステムの拡張性を考慮すると、機械学習による分類ができた方が良いため、別の料理にチューニングする場合も、大量のデータセットを取得する必要があり、その分の時間がかかると考えています。別の適した料理としては、とんかつや天ぷらなど、フライ調理により衣がつく料理は、適していると考えられます。

Dürst Martin 情報テクノロジー学科 教授

Q	Does the best timing for ending frying change depending on the type of coating of the fried chicken (e.g. mostly just spies vs. thick breading)?
A	ご質問ありがとうございます。本研究で利用したデータセットにおいても、パン粉を利用したものと、唐揚げ粉を利用した調理手法があり、その違いによる最適なタイミングは大きな違いがないことがわかっています。ただ極端なレシピにおいては、同様の特徴変化が捉えられない可能性があるため、最適なタイミングは変わることもあります。また、本研究では料理初心者向けのシステムであるため、そのような極端なカスタマイズは考慮しておりません。今後、個人の好みに合わせた最適な揚げ終わり検出が可能になるようシステムを拡張する場合、そのような極端なレシピも考慮する必要があるため、今後、検討したいと思います。

Dürst Martin 情報テクノロジー学科 教授

Q	What's the (average) overall cooking time for fried chicken?
A	ご質問ありがとうございます。インターネット上に掲載されている10種類のレシピを調査したところ、調理時間は平均4分程度となっており、本研究のデータセットにおいても、調理時間は4分程度となっております。

Dürst Martin 情報テクノロジー学科 教授

Q	Were you able to successfully use your system to fry delicious fried chicken?
A	ご質問ありがとうございます。本研究では、システムを利用してリアルタイムの性能評価を実施しておりません。また、本研究ではフライ調理が完了したかどうかを判定しており、美味しさについては考慮しておりません。今後、システムの評価を実施していきたいと考えております。また、ただ揚げ終わりかどうかを判定するだけでなく、美味しい揚げられているかも判定できるようにシステムを拡張していきたいと考えております。