

## 蕎麦の実自動選別装置の開発

– YOLOv7 と空気圧制御を用いたリアルタイム色選別手法の検討 –

鈴木 大貴\* 小高 知宏\* 黒岩 丈介\*

Development of an Automatic Buckwheat Grain Sorting System  
– A Study on Real-Time Color Sorting Using YOLOv7 and Pneumatic Control –

Daiki Suzuki\* Tomohiro Odaka\* Jousuke Kuroiwa\*

(Received January 1, 2026)

This study develops an automatic buckwheat grain sorting system using real-time image recognition and pneumatic control. A YOLOv7 model was optimized with ONNX and executed on a Raspberry Pi 5 to classify defective grains at high speed. The system removes detected grains through an air-jet mechanism synchronized with inference results. Experiments with small batches achieved nearly 100 % sorting accuracy, demonstrating the feasibility of low-cost, real-time automation for reducing manual labor in buckwheat processing.

**Key words** : Deep Learning, IoT, YOLOv7

### 1. はじめに

近年、飲食業界における深刻な人手不足が社会問題として顕在化しており、特に食材の処理や仕込みといったバックエンド業務において省人化・自動化の必要性が高まっている。中でも蕎麦店では、営業終了後に従業員が手作業で蕎麦の実を色（白／黒）ごとに選別しており、この工程は多大な時間と労力を要するため、長時間労働の一因となっている。

こうした課題に対し、本研究では深層学習による画像認識技術を活用し、蕎麦の実を自動的に分類・選別する選別装置の構築を目的とした。物体検出モデル「YOLOv7-tiny」<sup>[1]</sup>を用いて蕎麦の実の色をリアルタイムに判別し、判定結果に基づいてエアポンプを制御することで、物理的な選別を自動化した。さらに Raspberry Pi 5 上での軽量化・高速化を図ることで、店舗レベルで導入可能な低コスト・省スペースな自動選別システムの実現を目指した。

本論文では、第 2 章で本研究の目的と位置づけ、第 3 章で選別装置のハードウェア構成とシステム設計を

述べ、第 4 章で自動配列から判定・噴射に至る選別手法を示す。第 5 章では実験条件および評価指標を説明し、第 6 章で整列達成率・選別成功率・処理時間の結果を報告する。最後に、第 7 章で高速化要因や誤検知抑制の検討、スケーラビリティと今後の課題を含む考察とまとめを述べる。

### 2. 本研究の目的と位置づけ

近年、飲食業界では配膳ロボットや自動フライヤー、発券機など、ホール業務や調理工程の省人化が進展している。一方で、食材の前処理や仕込みといったバックエンド業務は依然として人手に依存しており、自動化が十分に進んでいない領域である。

特に蕎麦店は地方を中心に小規模店舗が多く、深刻な人手不足に陥りやすい。既存の食品工場向け色彩選別機は高価かつ大型であり、小規模店舗が導入するには現実的ではない。このため、低コスト・省スペースで店舗レベルに適用可能な自動選別技術のニーズは高い。

本研究は、このような背景を踏まえ、深層学習を用いた蕎麦の実（白粒・黒粒）のリアルタイム自動選別システム<sup>[2],[3]</sup>を、Raspberry Pi 5 を中心とした低価格構成で実現することを目的とする。物体検出モデル YOLOv7-

\*大学院工学研究科 知識社会基礎工学専攻

\*Fundamental Engineering for Knowledge-Based Society, Graduate School of Engineering

tiny による色判別と、エアポンプによる物理的な排出制御を組み合わせることで、店舗規模でも運用可能な選別装置の構築を目指した。

本研究は、AI・IoT 技術<sup>[2]</sup>による小規模飲食店の省人化に向けた具体例として位置づけられ、地域飲食業の労働負荷軽減および業務効率化に寄与することが期待される。

### 3. 選別装置の選別工程と構成

従来、蕎麦の実の白粒および黒粒の選別は、人の目視による判断と手作業によって行われてきた。この作業自体は単純なものであるが、一定量の蕎麦の実を処理するためには、長時間にわたって同じ作業を繰り返す必要があり、作業者にとって大きな負担となる。また、作業量が多い場合には、選別作業に多くの時間を要するという課題もある。

そこで本研究では、蕎麦の実の配列から判定、排出に至る一連の工程を機械的および情報処理的に自動化することを目的とし、蕎麦の実自動選別装置を構築した。

本章では、まず蕎麦の実の選別工程を段階的に整理したうえで、本研究で対象とした自動選別の基本的な流れを示す。さらに、それらの工程を実現するために設計した装置の構造および全体構成について述べる。

#### 3.1 選別工程の概要

現在、蕎麦の実の白粒および黒粒の選別は、人の目視と手作業によって行われている。この作業は長時間にわたる集中力を必要とし、作業者への身体的・精神的な負担が大きいだけでなく、作業効率および選別精度の面でも課題を有している。また、人による判断では作業者間でばらつきが生じる可能性があり、一定品質の維持が困難である。

本研究では、これらの課題を解決するために、蕎麦の実の選別工程を自動化し、人手を介さずに連続的な選別を可能とする装置の開発を目的とした。本装置では、蕎麦の実を一粒ずつ処理し、撮像および判定を行ったうえで白粒と黒粒を自動的に分類・排出するシステムを構築した。

蕎麦の実の選別工程は、大きく以下の5段階に分けられる。

1. 蕎麦の実の供給
2. 蕎麦の実の自動配列
3. 蕎麦の実の単粒排出
4. 撮像および判定

#### 5. エアポンプによる分類排出

本選別装置では、このうち(2)~(5)の工程を自動化対象とし、蕎麦の実を整列させ、1粒ずつ撮像・判定し、最終的に分離するまでの一連の流れを機械的に実現した。図1に、本研究で対象とした自動選別工程の全体構成を示す。

本装置は主に「配列台」「搬送路」「エアポンプ部」から構成されており、Raspberry Pi 5を用いてこれらすべての動作を統合的に制御する構成とした。エアポンプ部は左右に2基設置されており、それぞれ白粒用および黒粒用の排出口に対応している。

配列台は振動によって蕎麦の実を一方向に整列させる機構を備えており、その先に設けた搬送路を通じて粒が順次撮像点へと移動するように設計した。搬送路は滑らかなアルミ製の傾斜面で構成され、摩擦を低減することで粒が一定速度で自然落下するよう調整している。撮像点の直下にはエア噴射ノズルを配置し、判定結果に応じて対象粒を左右いずれかの排出口へ吹き分ける構造とした。

#### 3.2 選別方法

本節では、蕎麦の実の投入から排出に至るまでの一連の動作について述べる。

まず、選別対象となる蕎麦の実を無作為に配列台上へ投入する。無作為な配置状態を図2に示す。配列台にはスマートフォン筐体の振動機能を利用し、周期的な微振動を与えることで粒同士の位置関係を変化させる。この振動により、蕎麦の実は徐々に一方向へと整列し、最終的に一列状に並ぶ。整列後の様子を図3に示す。

整列した粒は搬送路の入口付近に集まり、重力によって順次撮像点へと移動する。搬送路の末端では蕎麦の実が一時的に滞留するが、配列台から伝わる微細な振動によって一定間隔で1粒ずつ自然に落下する。撮像点に落下した蕎麦の実の様子を図4に示す。

撮像点に到達した粒はカメラによって画像が取得され、その情報に基づいて白粒・黒粒の判定が行われる。判定結果に応じて、撮像点の直下に設置されたエアポンプが作動し、対象となる粒を左右いずれかの排出口へ吹き分ける。

このように、

「落下 → 撮像 → 判定 → 噴射」

の動作が連続的に繰り返されることで、蕎麦の実は1粒ずつ自動的に選別される。一連の動作はRaspberry Pi 5によってリアルタイムに制御されており、人手を介さずに安定した逐次選別が可能である。



図1 選別工程

以上が、本研究で開発した蕎麦の実自動選別装置の基本動作原理である。

### 3.3 装置構成

本選別装置のハードウェア構成を図5に示す。制御装置には Raspberry Pi 5 を使用し、撮像用カメラおよびエアポンプ制御部と接続することで全体の制御を行う。

カメラは撮像点に設置され、蕎麦の実が落下する瞬間の画像を取得する。取得した画像は Raspberry Pi 5 に送信され、判定処理の入力として使用される。

エアポンプの駆動にはモータードライバを介して制御を行う。左右2基のエアポンプはそれぞれ白粒用および黒粒用に対応しており、判定結果に応じて適切な側のみが作動する構成とした。電源と制御信号は分離して配線し、信号干渉や誤動作を防ぐよう配慮している。

これらの構成により、蕎麦の実の投入から判定・排出に至るまでの一連の工程をひとつの統合システムとして実現した。



図2 無作為に配置された蕎麦の実



図3 自動で配列された蕎麦の実



図4 撮像点に落下した蕎麦の実

## 4. 選別システムの設計と処理手法

第3章では、蕎麦の実自動選別装置の全体構成および、各工程における物理的な仕組みについて述べた。

本章では、それらの工程を実現するために構築した画像処理および制御システムの設計と処理手法について詳述する。特に、蕎麦の実の判定に用いた物体検出モデル、その推論をリアルタイムで実行するための最適化手法、および判定結果に基づいてエアポンプを制御する信号処理の方法について説明する。

本研究では、Raspberry Pi 5 を中核とした組み込みシステム上で、画像取得から分類、排出制御までを一貫して実行可能な構成を構築した。本章では、その設計思想と実装の詳細について順に述べる。

### 4.1 システム全体構成

本研究で構築した選別システムは、撮像部、判定部、制御部、および駆動部から構成される。図6に示すように、蕎麦の実が撮像点に到達するとカメラによって画像が取得され、その画像が物体検出モデルへ入力される。モデルによる判定結果に基づき、Raspberry Pi 5 がエアポンプの駆動信号を生成し、粒を左右いずれかの排出口へと吹き分ける。

本システムは、Raspberry Pi 5 を中心としてすべての

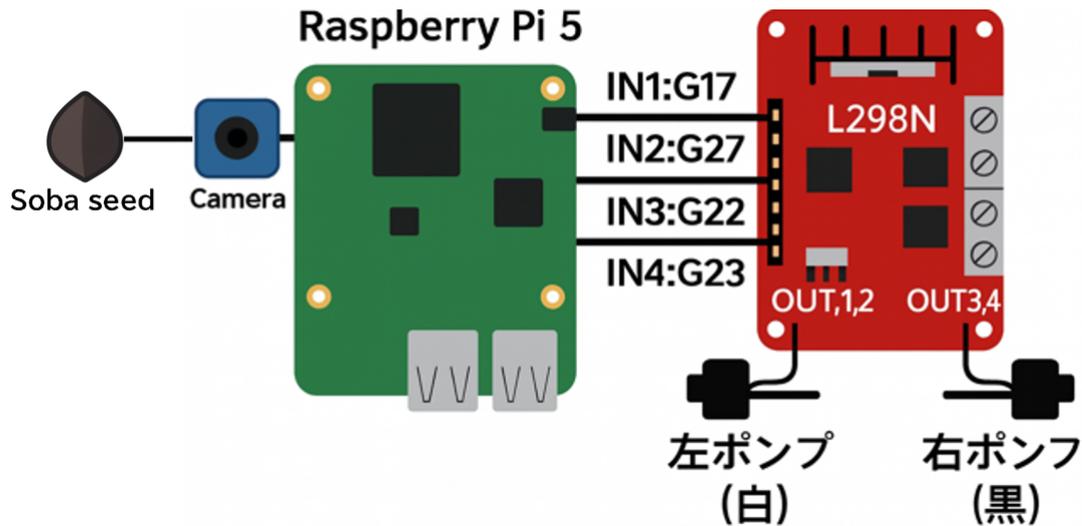


図5 ハードウェアの構成図

処理が統合されており、カメラによる撮像からエアポンプによる排出までがリアルタイムに制御される構成となっている。

#### 4.2 データセット構築

本研究では、蕎麦の実（白粒・黒粒）を対象とした物体検出モデルを構築するため、独自に小規模データセットを作成した。撮影には Raspberry Pi 5 Camera Module を使用し、以下の2種類の環境で画像を取得した。

- 単純背景（濃緑の平面上）：白粒2種・黒粒2種、各150枚
- 実機背景（アルミ搬送路+照明ムラ環境）：白粒1種・黒粒1種、各150枚

取得した画像は、全体を学習用70%、検証用20%、評価用10%の割合で分割した。本研究ではデータ拡張は行わず、異なる背景条件下におけるモデルの汎化性能を評価できる構成とした。

#### 4.3 撮像および前処理

撮像には Raspberry Pi 5 Camera Module を用い、蕎麦の実が撮像点に到達したタイミングでフレームを取得した。

取得した画像は BGR 形式で読み込まれた後、RGB 形式へ変換し、入力解像度を  $640 \times 640$  にリサイズした。さらに、画素値の正規化およびテンソル変換を行い、物体検出モデルに適した形式へ変換した。

一部のデバイスにおいてフレームサイズ情報が不安定となる事象が確認されたため、フレーム形状を固定

化する補正処理を実装し、安定した推論が行えるようにした。

#### 4.4 物体検出モデル

物体検出には YOLOv7-tiny を採用し、白粒および黒粒の2クラスの検出を行った。

YOLOv7-tiny は高い検出精度を保ちつつ軽量であり、組み込みデバイス上でのリアルタイム推論に適したモデルである。本研究では、自作データセットを用いて学習を行った。

学習済みモデルは、Raspberry Pi 5 上での高速な推論を実現するため、PyTorch 形式から ONNX 形式へ変換した。変換には YOLOv7 公式の `export.py` を使用し、OpenCV DNN との互換性を考慮して `--grid` オプションを有効化した。

変換後のモデルは Netron を用いて検証し、Conv + Batch Normalization の融合などの最適化が適用されていることを確認した。

#### 4.5 推論処理の最適化

推論には OpenCV の DNN モジュールを使用した。入力画像は `cv2.dnn.blobFromImage` を用いて、NCHW 形式および画素値範囲  $[0, 1]$  に正規化してモデルへ入力した。

また、I/O ボトルネックを軽減するため、V4L2 の仮想デバイスを介し、`libcamera-vid` および `ffmpeg` を用いた YUV420 から BGR24 への変換パイプラインを構築した。

これにより、入出力処理と推論処理の分離が可能となり、Raspberry Pi 5 の CPU キャッシュ効率およびス

ループットが向上した。

PyTorch 形式での推論時間は 1 フレームあたり約 9 秒であったのに対し、ONNX 形式に変換後は約 0.3 秒（約 3 FPS）となり、リアルタイム選別に十分な処理速度を達成した。表 1 に、モデル形式ごとの推論時間を示す。

#### 4.6 選別制御処理

推論結果に基づき、Raspberry Pi 5 から L298N モータードライバへ制御信号を送信し、エアポンプを駆動した。

GPIO 制御には gpiozero ライブラリを用い、白粒が検出された場合には左側ポンプ（GPIO17 / 27）を、黒粒が検出された場合には右側ポンプ（GPIO22 / 23）を作動させた。

ポンプの駆動時間は約 0.5 秒とし、短時間のエア噴射によって、蕎麦の実を確実に左右へ吹き分けるよう調整した。信号の誤動作を防止するため、制御信号と電源ラインは分離して配線した。

また、ポンプの連続動作による干渉を防止するため、推論完了後に一定のインターバルを設ける制御を実装した。

以上の設計により、蕎麦の実の画像取得、AI による分類判定、および空気圧による物理的な仕分けをひとつの統合システムとして実現した。

### 5. 選別処理の実験方法および評価指標

#### 5.1 実験条件

本実験は、提案装置の動作精度および処理性能を評価することを目的として実施した。白粒および黒粒の蕎麦の実を白粒と黒粒を各 10 粒（計 20 粒）用意し、混合した試料を用いた。実験はこれを 10 回計測する。図 2 に示すように、配列台上に蕎麦の実 20 粒をランダムに配置した試料について、撮像・推論・噴射・排出の一連動作を連続的に行わせた。

実験環境の構成を表 2 に示す。システムの中核には Raspberry Pi 5 を用い、ソフトウェア環境として OpenCV 4.11.0、ONNX 形式に変換した YOLOv7-tiny モデル、および gpiozero ライブラリを採用した。本構成により、撮像から制御までの全処理を単一デバイス上で完結させた。

各試料の通過時には、推論結果に基づき白粒を左側、黒粒を右側の排出口へエアポンプにより噴射した。全 20 粒の処理について、正しい選別数および 1 粒あたりの推論時間を記録した。

#### 5.2 評価指標

本実験では、自動選別装置の総合性能を以下の指標で評価した。

- 選別成功率（Accuracy）：単粒として排出され、かつ正しく選別された粒の割合。

$$\text{Acc} = \frac{\text{正しく選別}}{\text{単粒排出数}} \times 100[\%]$$

- 整列達成率（Alignment Rate）：配列台上で一列に整列し、単粒として撮像点に到達した粒の割合。

$$\text{AR} = \frac{\text{整列完了粒数}}{\text{投入粒数}} \times 100[\%]$$

- 平均推論時間（Inference Time）：撮像から推論完了までの平均処理時間。

これらの指標に基づき、ONNX 推論化による高速化効果と選別精度の両面から評価を行った。

### 6. 実験結果

表 3 に 10 回の実験結果を示す。全試行の平均値として、整列達成率は 82%、選別成功率は 99%、選別完了時間は 89.7 秒となった。各試行において、選別成功率はおおむね高い値を示した一方、整列達成率および完了時間には若干のばらつきが見られた。

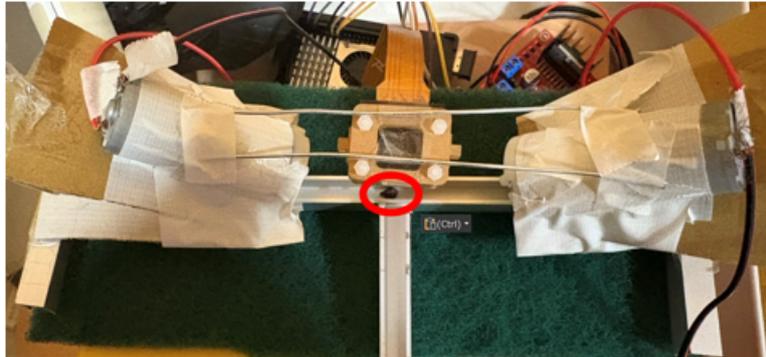
### 7. 考察とまとめ

#### 7.1 高速化の要因

PyTorch による直接推論では、CPU 上でのモデル展開および後処理（バウンディングボックス描画やスコア計算）に多大な計算資源を要し、1 フレームあたり約 9 秒を要した。そのため、リアルタイム動作は実現できなかった。一方、モデルを ONNX 形式へ変換し、OpenCV DNN モジュールを介して推論を行った結果、OpenCV の軽量なバックエンド最適化（Intel MKL による行列演算高速化および SIMD 命令の活用）が有効に機能し、推論時間は約 0.3 秒（約 3 FPS）まで短縮された。これにより、約 30 倍の高速化を実現した。

さらに、画像入力に関しては仮想カメラデバイスを用いた連続バッファ処理を導入した。これにより、画像取得と推論処理の非同期化が可能となり、I/O 待機時間が大幅に削減された。また、画像の前処理（リサイズ・正規化・色空間変換）をパイプライン化し、CPU キャッシュ効率を向上させたことも全体的なスループット向上に寄与したと考えられる。これら一連の最適化により、Raspberry Pi 5 単体でリアルタイム推論を実現

[撮像]

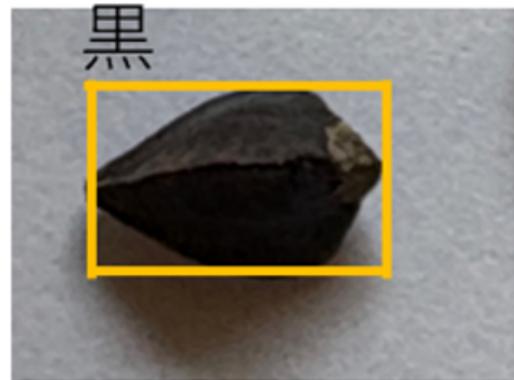
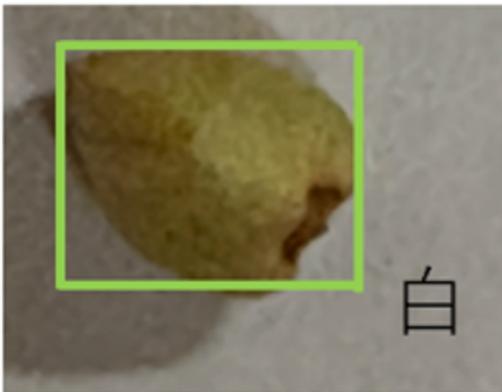


[YOLOv7-tiny (ONNX) で推論]

◇ 判定：白？黒？

はい (白)

いいえ (黒)



[左ノズル | 噴射 0.5 s]

[右ノズル | 噴射 0.5 s]

図 6 処理フロー

表 1 モデル別推論時間比較 (Raspberry Pi 5)

モデル形式	推論時間 [s/frame]	備考
YOLOv7 (通常版)	約 11.0	高精度モデル, 非リアルタイム
YOLOv7-tiny (PyTorch)	約 9.0	軽量版, 遅延あり
YOLOv7-tiny (ONNX)	約 0.3	OpenCV DNN, リアルタイム可

表 2 実験環境の構成

項目	内容
デバイス	Raspberry Pi 5
OS	Raspberry Pi OS (64-bit)
カメラ	USB 接続カメラ (640 × 480, 30 FPS)
モデル	YOLOv7-tiny (ONNX 形式)
ライブラリ	OpenCV 4.11.0, gpiozero
制御対象	エアポンプ (L298N 経由)
対象粒数	白粒 10 粒, 黒粒 10 粒 (計 20 粒)

し、外部 GPU に依存しない組込み AI システムとして実用的に機能することを確認した。

## 7.2 誤検知抑制

本研究では、誤検知の主因であった影や反射の影響を低減するために、撮像環境の背景色を濃緑に統一した。背景の統一によって粒子の外縁が明確に分離され、学習モデルが影や光のムラを物体として誤認識するリスクを抑制した。

さらに、学習データセットの構築段階においては、照明条件や撮影角度を意図的に変化させ、さまざまな形状・方向の影を含む画像を多数取り入れた。これにより、モデルが影を「特徴」としてではなく「背景の一部」として学習するよう調整を行った。その結果、実際の選別環境において発生する照明装置やアルミフレームによる陰影、カメラ設置位置の微小なズレに起因する影なども誤検知されなくなった。

これらの環境的およびデータの改善が複合的に作用した結果、モデルの汎化性能が向上した。その結果、平均 16 粒中 15.9 粒 (99%) を正確に分類・排出することができた。したがって、影の抑制とデータ拡張を組み合わせた対策が、選別精度 99% 達成の主要因であったと考えられる。

一方で、黒色の蕎麦の実のうち角が立っているものが単粒として排出された後、アルミフレームに衝突し、カメラ枠外に飛散する現象が確認された。この課題に対しては、現行の搬送路傾斜 (20°) を 15° へ緩和する、あるいは搬送路の長さを短縮するなど、設計変更による検証が必要であると考えられる。

## 7.3 自動整列の安定性とスケーラビリティ

本研究では、20 粒規模の条件下において配列台上での一列整列を実施した結果、整列達成率は 82% であった。100% に至らなかった要因としては、一列に並んだ蕎麦の実を後続の粒が押し出す現象が確認された。これに対して、将来的には粒が配列部から落下しないように壁を設けることが有効と考えられる。ただし、壁の追加により摩擦が増加し、蕎麦の実が詰まる懸念もあるため、摩擦係数の低い素材 (例: テフロン樹脂やアクリル素材) の採用や、壁面を内側に約 10° 傾斜させる設計が有効と考えられる。

また、100 粒規模の連続選別を想定した場合、配列密度の上昇に伴い粒同士の接触や押し合いが発生し、押し出しや転落が生じる可能性がある。この課題に対しては、流路入口部に仕切り構造を設け、粒を単列で安定的に搬送する設計が有効と考えられる。仕切りによって流路を複数に分割し、各経路を単列通過に限定することで、過剰供給を抑え、単粒排出を安定化できると期待される。

今後は、摩擦係数の異なる素材の比較実験や流路形状・角度・表面処理の最適化を行い、詰まりや脱落の発生率を定量的に評価する予定である。特に、「粒の密集防止」「滑らかな流動性の確保」「単粒排出制御の安定化」という三つの観点から設計を進め、100 粒規模での連続選別を実現することを目指す。

## 7.4 制約と展望

本研究で構築した選別装置は 20 粒規模での実験を対象としていたが、今後は 100 粒規模、さらに 200 粒規模への拡張を目指す。粒数の増加に伴い、単粒ごとの整列・排出をより精密に制御する必要がある。特に、配列台上に多数の蕎麦の実が一度に供給される場合、粒が重なり合ったり溢れ出したりする可能性が高く、現行の構造では単粒の抽出が困難となる。

これを解決するために、今後は配列台の構造を再設計し、「あふれた蕎麦の実の中から 1 粒を確実に排出する」ための新たな仕組みを検討する。具体的には、配列工程を二段階化し、まず粒を一列に整列させた後、排出口手前で入口幅を徐々に狭めることで、過剰な粒を自然に選り分ける方式が有効と考えられる。このよう

表3 自動配列および選別結果（10 試行）

試行回	自動配列数（割合）	選別成功率（割合）	選別完了時間 [s]
1	16/20（80%）	16/16（100%）	63
2	18/20（90%）	18/18（100%）	90
3	15/20（75%）	15/15（100%）	82
4	16/20（80%）	15/16（94%）	104
5	16/20（80%）	16/16（100%）	82
6	19/20（95%）	18/19（95%）	81
7	15/20（75%）	15/15（100%）	85
8	16/20（80%）	16/16（100%）	92
9	18/20（90%）	18/18（100%）	110
10	15/20（75%）	14/15（93%）	88
平均	16.4/20（82%）	15.9/16（99%）	89.7

な段階的狭窄構造により、重なりを防ぎつつ一定速度で粒を送り出すことが期待できる。

さらに、200 粒規模の処理を実現するためには、単一の配列台では供給密度と選別速度の両立が困難となる。そのため、複数の配列台を備えた供給台ユニットを新たに設計し、それぞれが独立して粒を整列・排出できる構造を導入する予定である。これにより、複数レーンの並列処理が可能となり、大規模処理への展開が現実的となる。

最終的には、これらの改良を通じて、蕎麦の実を 100 粒・200 粒単位で連続的に選別できる高スループット装置を構築し、飲食店や製造現場での省人化・自動化を実現することを目指す。

#### 参考文献

- [1] WongKinYiu, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," GitHub, 2022.
- [2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016.
- [3] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.
- [4] ONNX, "Open Neural Network Exchange," <https://onnx.ai/> (accessed 2025-11-16).
- [5] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.