平成 14 年度

学士学位論文

医療用食事摂取量計測システムにおける
食器抽出処理の開発に関する研究

Research on Development of Dishes Extraction for Food Intake Measuring System

1030305　藤原浩一

指導教員：竹田史章教授

2003 年 2 月 12 日

高知工科大学工学部情報システム工学科

要旨

医療用食事摂取量計測システムにおける
食器抽出処理の開発に関する研究

藤原浩一

著者は，これまで医療用食事摂取量計測システムの開発を行ってきた．このシステムにおいて，実際の食事摂取量計測はトレイに配置されている食器毎の食材を基に摂取量を計測する．そのため，システムにおいては食器を特定し抽出する必要がある．そこで，本研究ではこの食器を抽出する処理にニューラルネットワーク（以下 NN と略記）を用いることを提案する．NN を用いることにより各食器同士がある程度重なりあっている状態においても食器を高速かつ正確に抽出することが期待される．本研究では NN を用いた食器抽出処理機構を開発し，食器抽出実験を行い有用性を実データを用いたシミュレーションで示す．

キーワード　：食事摂取，ニューラルネットワーク，食器抽出

Abstract

Research on Development of Dishes Extraction for Food Intake Measuring System

Koichi FUJIWARA

We have developed food intake measuring system. This system measures amounts of food intake for each dish on the tray. Therefore, this system needs to extract dish from a tray each by each. In this paper, we propose a dish extraction algorithm by neural network (NN). We expect that the proposed method can exactly extract dish image even if dishes are overlapped each other to extract over dishes exactly. In this paper, we develop the method of dish extraction by the NN. We show possibility and basic ability with computer simulation with real data.

Keyword: Neural Network , Food intake, Extraction

目次

1. はじめに……1

2. 食事摂取量計測システム……3

2.1 開発背景……3

2.2 これまでの研究……6

2.2.1 食材抽出実験……6

2.2.2 NN による識別実験……7

2.2.3 これまでの問題点……12

3. 食事摂取量計測システムの構成……13

3.1 システム構成……13

3.1.1 計測部の構成……14

3.1.2 データベース部の構成……15

3.1.2.1 食材データベース……16

3.1.2.2 料理データベース……17

3.1.2.3 食器データベース……18

3.1.2.4 盛付けデータベース……19

3.1.2.5 メニュー内容データベース……20

3.1.2.6 メニュー提供者データベース……21

3.1.2.7 個人データベース……22

3.1.2.8 個人摂取履歴……23

3.2 動作フロー……24

4. NN による食器抽出処理……26

4.1 食器抽出手法……26

4.2 動作フロー……26

4.2.1 画像の 2 値化……..27
4.2.2 NN による判定……28
4.3 食器抽出用 NN の構造……30
5. 食器抽出実験……31
5.1 食器抽出実験 1……..31
5.1.1 実験条件……..31
5.1.2 実験結果……...36
5.2 食器抽出実験 2……..40
5.2.1 実験条件……..40
5.2.2 実験結果……..44
5.3 考察……46
6. 食物除去アルゴリズム……47
6.1 アルゴリズムの概要……47
6.2 動作フロー……49
7. 食物除去アルゴリズムの性能確認実験……50
7.1 実験条件……50
7.2 実験結果……50
7.3 考察……..55
8. まとめ……57
9. 謝辞……58
10. 参考文献……59
付録……63
ニューラルネットワーク……63
食物除去アルゴリズムを反映させた食器抽出ソースコード……69

図目次

図 2.1　配膳風景……..3

図 2.2　アンケート内容……...4

図 2.3　アンケート結果……...5

図 2.4　食材抽出結果……...7

図 2.5　学習画面……...8

図 3.1　システム構成図……13

図 3.2　システム概観図……14

図 3.3.　メインシステム動作画面……15

図 3.4　食材データベース……16

図 3.5　料理データベース……17

図 3.6　食器データベース……18

図 3.7　盛付けデータベース……19

図 3.8　メニュー内容データベース……20

図 3.9　メニュー提供者データベース……21

図 3.10　個人データベース……..22

図 3.11　個人摂取履歴……..24

図 3.12　動作フロー図……..25

図 4.1　NN による食器抽出処理……26

図 4.2　動作フロー図……27

図 4.3　2 値化画像……27

図 4.4　データベースシステム……28

図 4.5　切り出し画像の入力……29

図 4.6　食器抽出用 NN の構造……30

図 5.1　**使用食器画像**……31

図 5.2　**食器配膳例**……32

図 5.3　**食器位置特定成功例**……37

図 5.4　**食器位置特定失敗例**……37

図 5.5　**食器の配置**……40

図 5.6　**実験データ（**1**）：残量** 0%……41

図 5.7　**実験データ（**1**）：残量** 20%……41

図 5.8　**実験データ（**1**）：残量** 40%……41

図 5.9　**実験データ（**1**）：残量** 60%……41

図 5.10　**実験データ（**1**）：残量** 80%……42

図 5.11　**実験データ（**1**）：残量** 100%……42

図 5.12　**実験データ（**2**）：残量** 0%……42

図 5.13　**実験データ（**2**）：残量** 20%……42

図 5.14　**実験データ（**2**）：残量** 40%……43

図 5.15　**実験データ（**2**）：残量** 60%……43

図 5.16　**実験データ（**2**）：残量** 80%……43

図 5.17　**実験データ（**2**）：残量** 100%……43

図 5.18　**抽出成功例**……45

図 5.19　**抽出失敗例**……45

図 6.1　NN **入力画像（**1**）**……47

図 6.2　NN **入力画像（**2**）**……47

図 6.3　**食物除去アルゴリズム**……48

図 6.4　**動作フロー**……49

図 7.1　**抽出成功例**……54

図 7.2　**抽出失敗例**……54

図 7.3　**実験結果（プログラム変更後）**……55

図 7.4　**実験結果（プログラム変更前）**……55

図 7.5　NN **入力画像（プログラム変更後）**……56

図 7.6　NN **入力画像（プログラム変更前）**……56

表目次

表 2.1　残量 0%画像の識別結果……...9

表 2.2　残量 20%画像の識別結果……9

表 2.3　残量 40%画像の識別結果……10

表 2.4　残量 60%画像の識別結果……10

表 2.5　残量 80%画像の識別結果……11

表 2.6　残量 100%画像の識別結果……..11

表 5.1　NN の学習および評価条件……..32

表 5.2　食器情報……33

表 5.3　メニュー一覧（1）……..34

表 5.4　メニュー一覧（2）……..35

表 5.5　実験結果（1）……..38

表 5.6　実験結果（2）……..39

表 5.7　実験結果（1）……..44

表 5.8　実験結果（2）……..44

表 7.1　実験結果（1）……..51

表 7.2　実験結果（2）……..52

表 7.3　食器位置特定率（1）……..53

表 7.4　食器位置特定率（2）……..53

第 1 章　はじめに

これまで，著者は人の手に代わり機械による食事摂取量計測を実現するため，食事摂取量計測システムの開発を行ってきた．開発背景として，病院内における入院患者の食事摂取量計測方法の問題がある．病院内における食事摂取量計測方法の現状は，入院患者の摂取した食事を栄養士が目で確認を行い，食品成分表と照らし合わせて食膳全体の摂取栄養素を計測する方法が採用されている．そのため，計測基準が各栄養士によって異なり，計測結果が曖昧である．また，入院患者数などを考慮すると，膨大な計測時間を費す．さらに，計測したデータが電子化されていないため，データ管理が困難である．以上の点より，機械による食事摂取量計測を実現するために，食事摂取量計測システムの開発を行ってきた[(1)-(6)]．食事摂取量計測システムの基本コンセプトは食事後の食膳画像から摂取量計測を行うことである．

開発システムでは，各食器毎に食事摂取量の計測を行う必要がある．このため，計測の前処理として，トレイ上のどの位置に食器があるかを特定する必要がある．しかしながら，開発中のシステムにはこの食器抽出機構が組み込まれていない．そこで，本研究では，ニューラルネットワーク（NN と略記）を用いた食器抽出処理の開発を行う[(3)-(6)]．NN を用いる理由[(7)-(26)]は，通常の画像処理[(27)-(29)]によって全画素を探索するのに比べて処理時間が短縮されることや，食器から食物がはみ出しているような曖昧な画像にも対応するためである．

食器抽出処理流れとしては，まず，食膳画像を食器毎の適したしきい値によって 2 値化を行う．次に，2 値化画像から食器大の画像を切り出し，NN に入力する．NN による判定を行い，NN 反応値が最も高いとされたものを食器を含んだ画像として抽出を行う．本研究では，前述のシステムを用いて食器抽出処理機構の機能を測るための食器抽出実験を行う．

食器抽出実験の結果より食器抽出を行う際，問題として挙げられる点のひとつに，

食器上に食物が載っている場合，それがノイズとなってしまい，正しい画像が NN **に入力されないということが挙げられる．そこで，本研究では，**NN **に食膳画像を入力する際に入力画像から食物部分の除去を行うアルゴリズムの提案，及び，プログラムの改良，改良後のシステムを用いた食器抽出実験を行うことで，現在の食事摂取量計測システムにおける食器抽出機構の機能向上を図る．**

第 2 章　食事摂取量計測システム

2.1　開発背景

現在，病院内における入院患者の食事摂取量計測は，栄養士が残食材を目視により，食事摂取量の計測を行い，食品成分表と照らし合わせながらカロリーの摂取量を算出する方法が用いられている．しかしながら，現状の計測方法ではいくつかの問題がある．具体的な問題点として，まず，摂取量計測の判断基準が各栄養士によって異なるため，測定基準が一定にならない点．入院患者数が多くなった場合，計測に膨大な時間がかかってしまうという点．さらに，計測結果などの電子化が行われていないため，データ参照やデータ管理等が不便であるという点である．開発システムは，今まで栄養士が行っていたこれらの作業を機械によって行い，先に示してきた現状の病院内における入院患者の食事摂取量計測に対する問題を解決するということを目的とする．図 2.1 に現在の病院内の配膳風景を示す．

図 2.1　配膳風景

本システムの開発目的として, 栄養士をはじめとする病院関係者がこのシステムの開発を強く望んでいるということがある．それを実証するために，高知県内の給食事業に従事する病院関係者を対象に図 2.2 のようなアンケートを実施した.

病院内において患者個人のエネルギー及び栄養素摂取量を正確に把握することは，栄養管理上非常に重要であるが，聞き取りによる残食料の調査等は人手の関係で大変です．そこで，短時間に各人の摂取量を検査するシステムがあればと考えます．高知工科大学との共同研究により，人手がかからず，短時間に給食の摂取量を検査し, エネルギーや栄養素の摂取量を正確に算出するシステムの開発に取り組みたいと考えます．これまでの研究では，カメラを内臓した画像認識装置を応用して, 食事前後のトレイ上の食物の差からエネルギーや栄養素の摂取量を算出するシステムの開発が可能であると思われます. そこで給食事業に携わっている先生方のご意見をお聞かせ願いたく, アンケート調査をさせていただきます.

質問 1. このようなシステムについてどう思いますか？
（1）必要性を感じる
（2）必要性を感じない
（3）考えていなかった

質問 2. このようなシステムが開発されれば，導入したいと思いますか？
（1）導入したい
（2）興味がある
（3）検討してみた
（4）興味がない

質問 3. 導入できる価格はどの範囲ですか？
（1）10〜20 万円
（2）20〜30 万円
（3）30〜50 万円

図 2.2　アンケート内容

図 2.2 のアンケートに対し，栄養士 55 名，医師 1 名，看護婦 1 名の計 57 名の方から回答を得た．回答の集計結果は図 2.3 のとおりである．

質問１の回答では,
（1）必要性を感じる…26 人（46%）
（2）必要性を感じない…２人（4%）
（3）考えていなかった…29 人（50%）

質問１で必要性を感じると答えた人で，質問２では,
（1）導入したい…5 人（9%）
（2）興味がある…16 人（28%）
（3）検討してみたい…5 人（9%）
（4）興味がない…０人

質問１で考えていなかったと答えた人で，質問２は,
（1）導入したい…０人
（2）興味がある…12 人（21%）
（3）検討してみたい…5 人（9%）
（4）興味がない…4 人（7%）
無記入…8 人（14%）

質問 3 の回答では,
（1）10〜20 万円…33 人（58%）
（2）20〜30 万円…7 人（12%）
（3）30〜50 万円…0 人
分からない，無記入…17 人（30%）

図 2.3　アンケート結果

アンケートの結果, 必要性を感じる人が 57 人中 26 人で 46%と半数に満たないが,「考えていなかった」と回答した 29 人のうち 17 人は「興味がある」「検討してみたい」と回答した．この結果，57 人のうち 43 人（75%）が本システムに関心をよせていることが分かる．

2.2　これまでの研究

これまで著者は，食事摂取量計測システムの開発を行ってきた[(1)-(6)]．本節ではこれまで行ってきた研究について述べる．

2.2.1　食材抽出実験

開発システムでは，画像処理で食器上の食材を抽出する処理を行う．食材の抽出処理のアルゴリズムを以下に示す．

256 色ビットマップファイルから食膳画像を取り出し，RGB の内，青色画像だけを取り出す．次に，しきい値を用いて画像を 2 値化する．

(a)　しきい値以上の画素を”1”画素とする．

(b)　しきい値以上より小さい画素を”0”画素とする．

ここで，”1”画素は食材の部分で”0”画素は他の背景部分になる．次に，画像内のノイズを除去するために 8 近傍収縮処理を行う．収縮処理に関しては，収縮回数を 1 回から 100 回までの内から任意に決めることが可能であり, 実験にて試行錯誤的に最適な収縮回数を 4 としている．8 近傍収縮処理の後に，以下のラベル処理行う．

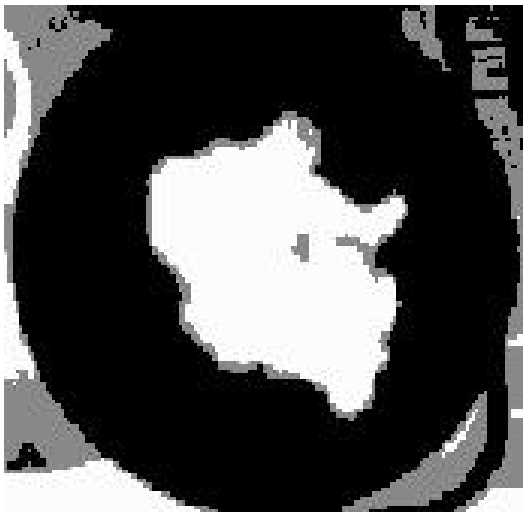
(a)　“1”画素を探索する．

(b)　探索した”1”画素に，現在のラベル番号を付ける．

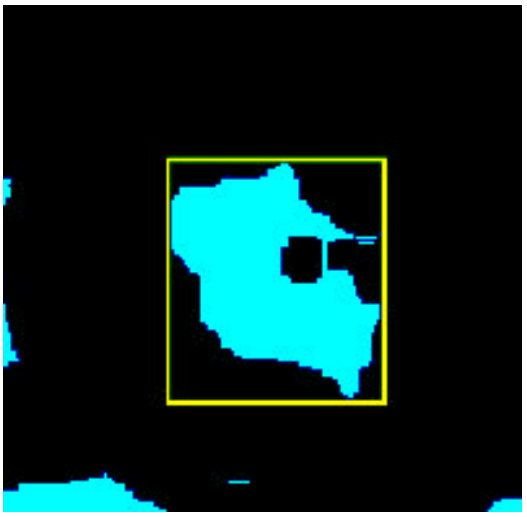
(c)　現在のラベル番号の 8 近傍に”1”画素があれば，その画素に同じラベル番号を設定する．

その後，各々のラベル領域の中心位置を求める．最後に，求めたラベル領域ごとに，領域の位置の最大値，最小値を求める．

この抽出処理を使用して食器画像から食材画像を抽出する実験を行った．図 2.4 に食材の抽出処理結果を示す．



（a）対象画像



（b）抽出結果

図 2.4　食材抽出結果

2.2.2　NN による識別実験

本項では食材残量パターンを NN によって識別する手法について記載する．使用する NN の構成は，3 層階層型 NN である．まず，食材画像を判定し，摂取カロリーを算出する．具体的なアルゴリズムを以下に示す．

(a)　典型的な食事摂取量をレベル分類（例えば，Level-1=0%，Level-2=20%，Level-3=40%，…，Level-6=100%）し，食材残量 100%に対するカロリーAcal を算出する．

(b)　食後の残食材を皿毎に抽出し，残食レベルを確定する．

(c)　皿毎の摂取カロリーa は(1)式で算出する．

$$a\ cal = \{1 - (\text{確定レベルの}\%) / 100\} \times A\ cal \qquad (1)$$

このアルゴリズムを基に，食材残量識別実験を行った．食材は炊飯米を使用し，25W のくもりガラス球，拡散板を撮影筐体に設置し，トレイはピンク色トレイ，茶碗は黒いプラスティック製の茶碗を使用する．炊飯米の分類パターンは，残量がそれぞれ 0%，20%，40%，60%，80%，100%の 6 パターンとする．1 つのパターンには食器の回転や位置ずれを考慮し，トレイ上の 25 の配置に食器を回転させながら置き直し，

撮影した画像を使用する．実験は，学習用データ 5 枚に中央付近の 5 箇所配置したものを使用し行う．なお，学習用，評価用データに用いた画像データはマニュアルで食器部分を抽出する．学習画面を図 2.5 に示す．実験結果を表 2.1～表 2.6 に示す．評価用データは学習用データの 5 枚を除いた 20 枚の画像データとする．画像データ名である riceX-Y の X は炊飯米の残量パターンを，Y は 25 通りの食器配置パターンを表す．表内の“○”は，評価用データの残量パターンと，NN により出力された各パターンの出力値のうち最も高い出力値を出力したパターンとが一致したことを表す．“△”は，評価用データの残量パターンと出力値の最も高いパターンとが一致したが，出力値が 0.5 に達していない状態を表す．“×”は，評価用データの残量パターンと出力値の最も高いパターンが一致していないことを表す．

実験の結果，残量 0%の識別率＝100%，残量 20%の識別率＝90%，残量 40%の識別率＝90%，残量 60%の識別率＝90%，残量 80%の識別率＝95%，残量 100%の識別率＝100%であった．全体識別率としては 94%を示した．

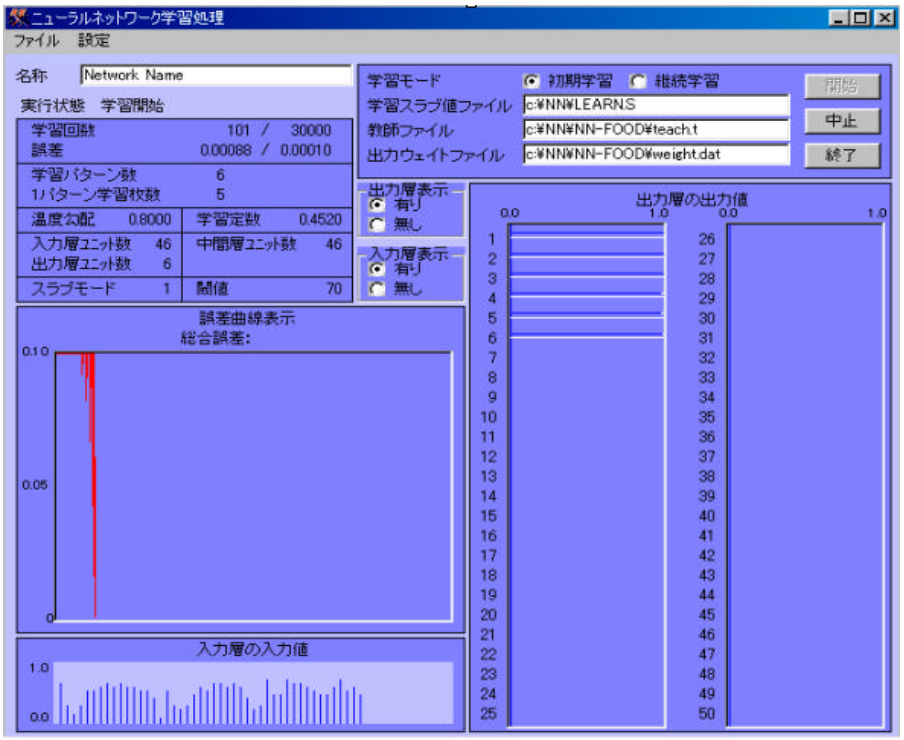


図 2.5　学習画面

表 2.1 残量 0%画像の識別結果

画像データ	識別結果
rice0-1	○
rice0-2	○
rice0-3	○
rice0-4	○
rice0-5	○
rice0-6	○
rice0-7	○
rice0-8	○
rice0-9	○
rice0-10	○
rice0-11	○
rice0-12	○
rice0-13	○
rice0-14	○
rice0-15	○
rice0-16	○
rice0-17	○
rice0-18	○
rice0-19	○
rice0-20	○
rice0-21	○
rice0-22	○
rice0-23	○
rice0-24	○
rice0-25	○

表 2.2 残量 20%画像の識別結果

画像データ	識別結果
rice20-1	○
rice20-2	○
rice20-3	○
rice20-4	○
rice20-5	○
rice20-6	○
rice20-7	○
rice20-8	○
rice20-9	○
rice20-10	○
rice20-11	○
rice20-12	○
rice20-13	○
rice20-14	○
rice20-15	○
rice20-16	○
rice20-17	○
rice20-18	○
rice20-19	○
rice20-20	○
rice20-21	○
rice20-22	○
rice20-23	×
rice20-24	×
rice20-25	○

表 2.3 残量 40%画像の識別結果

画像データ	識別結果
rice40-1	○
rice40-2	○
rice40-3	○
rice40-4	△
rice40-5	○
rice40-6	○
rice40-7	○
rice40-8	○
rice40-9	○
rice40-10	○
rice40-11	○
rice40-12	×
rice40-13	○
rice40-14	○
rice40-15	○
rice40-16	○
rice40-17	○
rice40-18	○
rice40-19	×
rice40-20	○
rice40-21	○
rice40-22	○
rice40-23	○
rice40-24	○
rice40-25	○

表 2.4 残量 60%画像の識別結果

画像データ	識別結果
rice60-1	○
rice60-2	○
rice60-3	○
rice60-4	○
rice60-5	○
rice60-6	○
rice60-7	○
rice60-8	△
rice60-9	×
rice60-10	○
rice60-11	○
rice60-12	○
rice60-13	○
rice60-14	○
rice60-15	○
rice60-16	○
rice60-17	○
rice60-18	○
rice60-19	○
rice60-20	×
rice60-21	○
rice60-22	○
rice60-23	○
rice60-24	○
rice60-25	○

表 2.5 残量 80%画像の識別結果

画像データ	識別結果
rice80-1	○
rice80-2	○
rice80-3	○
rice80-4	○
rice80-5	○
rice80-6	×
rice80-7	○
rice80-8	○
rice80-9	○
rice80-10	○
rice80-11	○
rice80-12	○
rice80-13	○
rice80-14	○
rice80-15	○
rice80-16	○
rice80-17	○
rice80-18	○
rice80-19	○
rice80-20	○
rice80-21	○
rice80-22	○
rice80-23	○
rice80-24	○
rice80-25	○

表 2.6 残量 100%画像の識別結果

画像データ	識別結果
rice100-1	○
rice100-2	○
rice100-3	○
rice100-4	○
rice100-5	○
rice100-6	○
rice100-7	○
rice100-8	○
rice100-9	○
rice100-10	○
rice100-11	○
rice100-12	○
rice100-13	○
rice100-14	○
rice100-15	○
rice100-16	○
rice100-17	○
rice100-18	○
rice100-19	○
rice100-20	○
rice100-21	○
rice100-22	○
rice100-23	○
rice100-24	○
rice100-25	○

2.2.3　これまでの問題点

これまで，食事摂取量計測用に開発したニューロ学習・識別システムを使用し，炊飯米の 6 パターンの残量パターンにおける食材残量識別実験において高い識別性能を得ている．しかしながら，実際に起こりうる無限の食材残量パターン数を考慮すると，NN 構成が困難となる．そのため，NN によって残食材を識別する手法に代わり，新しい摂取量計測手法を検討する必要がある．

また，計測対象となる食材を抽出する際，これまでは手動にて食材を含む食器画像の抽出を行っていた．このため，統合システムを開発する上で自動で食器抽出処理を行う必要がある．

第 3 章　食事摂取量計測システムの構成

2.2.3 で述べた問題点を改善するために，食器抽出機構を含めた，食事摂取量計測システムの構築を考案する．

3.1　システム構成

システム構成を以下の図 3.1 に示す．システム概観を図 3.2 に示す．撮影部分は，撮影筐体に取り付けられた USB カメラを摂取量計測を行う PC に接続したもので，摂取量計測を行う本体部分は，内部に摂取量計測システムを組み込んだ PC を用いる．トレイには RFID チップを取りつけ，個人に対応する食膳の自動判定を行う．

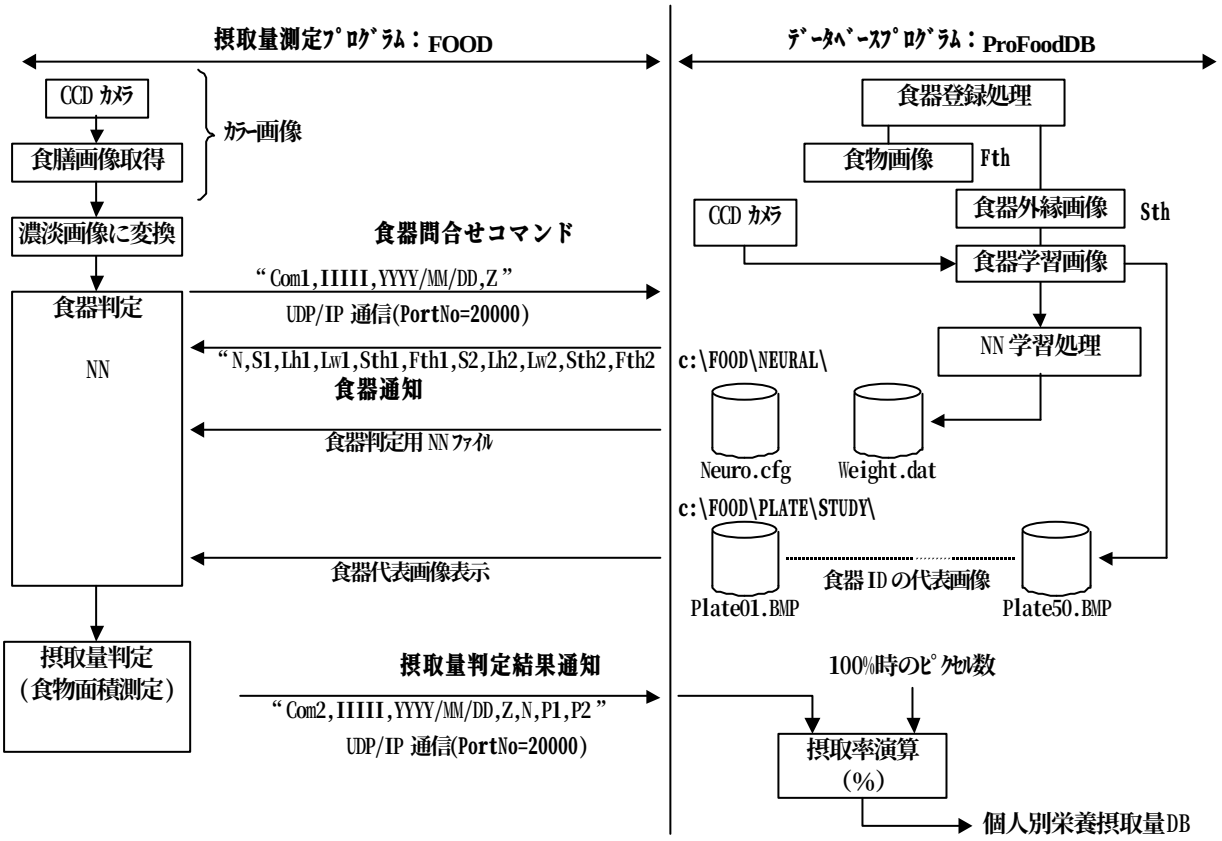


図 3.1　システム構成図

図 3.2　システム外観図

PC 内で実際に食事摂取量計測を行うシステムには大きく分類し 2 つのシステムがある．「計測部」と「データベース部」である．

3.1.1　計測部の構成

計測部では食器抽出や食材残量判定，摂取カロリー量の測定を行う．食器抽出処理はデータベース部から食器情報を取得し，NN による判定によって食器抽出を行う．食材残量判定は，食器抽出を行った後，食材部分の面積（画素数）から残量を測定する．摂取カロリー量の測定は，食前状態の画像との比較によって行う．摂取カロリー量を算出する式を(3.1)に示す．食材残量 100%時のカロリーを A cal，食材面積を S とする．食後のカロリーを a cal，食材面積を s とする．

$$a\ cal = \{\ 1\ -\ s / S\ \} \times\ A\ cal \qquad (3.1)$$

図 3.3 に動作画面の一部を示す．

図 3.3　メインシステム動作画面

3.1.2　データベース部の構成

食器情報などのデータは，データベース部に格納する．開発システムにおいて使用するデータは食材や食器，メニューや患者のデータなどである．また，患者ごとの摂取量などを登録する機能もある．

3.1.2.1　食材データベース

計測対象となる各食材の成分などは食材データベースに格納されている．この成分データは文部科学省資源調査会編「五訂日本食品標準成分」から引用する．以下に食材データベースに格納されているデータを示す．図 3.4 に食材データベースの画面を示す．

- 食材 ID
- 食品名
- 食品コード
- 成分値（ビタミン，カルシウムの成分値を格納）

図 3.4　食材データベース

3.1.2.2　料理データベース

料理データベースでは，調理された料理に関するデータを格納する．料理に関するデータとは，料理種別，料理名，使用食材などである．以下に料理データベースに格納されているデータを示す．図 3.5 に料理データベースの画面を示す．

- 料理 ID
- 料理種別
- 使用食材数
- 料理名
- 登録オプション（食材選んで登録か栄養成分を指定して登録から選択）
- 料理画像ファイル名
- 使用食品名
- 使用食品の使用量

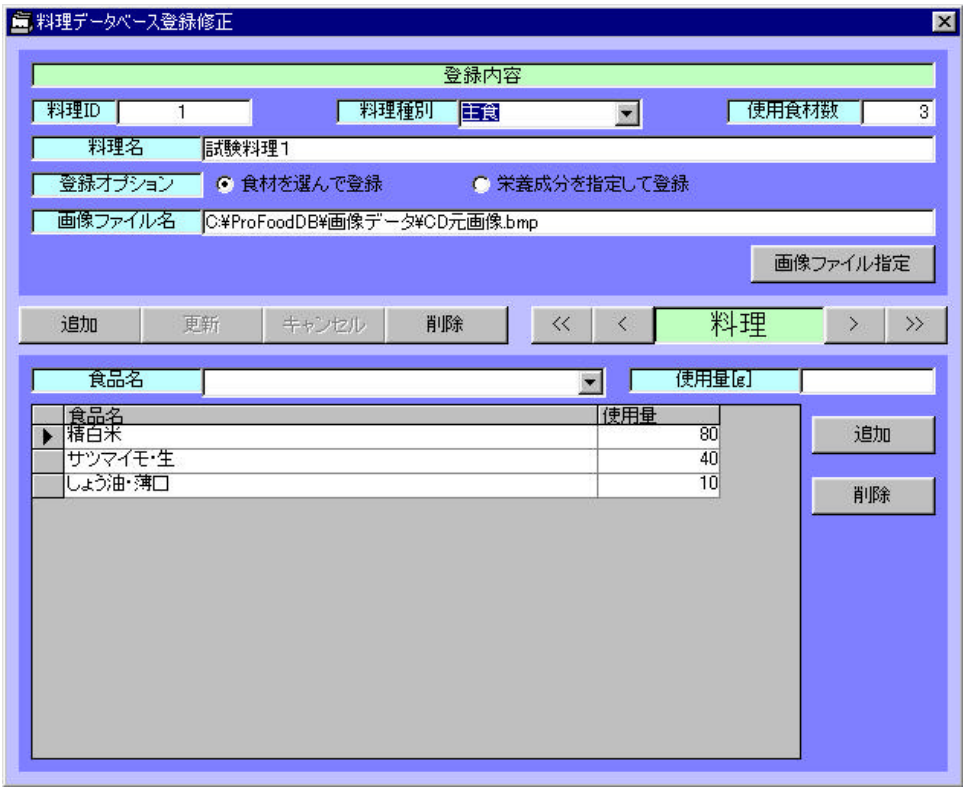


図 3.5　料理データベース

3.1.2.3　食器データベース

食器データベースは 3.1.1 で記述した食器位置特定処理の際に用いる．食器位置特定処理プログラムとは UDP/IP (User Datagram Protocol / Internet Protocol) 通信で通信を行う．食器データベースには食器位置特定処理に必要な食器の縦・横寸法，食器形状などが格納されている．食器データベースに格納されているデータを以下に示す．図 3.6 に食器データベース画面を示す．

- 食器 ID
- 形状（円形と方形から選択）
- 食器名
- 横方向寸法
- 縦方向寸法
- しきい値（2 値化の際に使用）
- 元画像ファイル名
- 画像縮小率
- 2 値画像ファイル名

図 3.6　食器データベース

3.1.2.4　盛付けデータベース

盛付けデータベースには調理された料理と盛付けに使用する食器のデータを格納する．この盛付けデータベースには食前の食物面積（画素）も格納されている．摂取量計測処理の際にこのデータを使用する．盛付けデータベースに格納されているデータを以下に示す．図 3.7 に盛付けデータベースの画面を示す．

- 盛付け ID
- 盛付け名
- 料理名
- 使用食器名
- 食物面積
- スレシホールド値
- 面積縮小率
- 元画像ファイル名
- 2 値画像ファイル名

図 3.7　盛付けデータベース

3.1.2.5　メニュー内容データベース

メニュー内容データベースでは，盛付けに使用する食器の数やその種類などを格納する．また，朝食や昼食などといった提供種別，提供年月日なども格納されている．メニュー内容データベースに格納されているデータを以下に示す．図 3.8 にメニュー内容データベースの画面を示す．

- メニューID
- メニュー名
- 提供年月日
- 提供種別（朝食，昼食，夕食）
- 提供数
- 盛付け数

図 3.8　メニュー内容データベース

3.1.2.6　メニュー提供者データベース

メニュー提供者データベースでは, 患者に食事を提供する栄養師に関するデータを格納する．格納するデータとしてメニュー名や提供種別（朝食，昼食，夕食）などがある．図 3.9 にメニュー提供者データベース画面を示す．以下にメニュー提供者データベースに格納されているデータを以下に示す．

- メニューID
- メニュー名
- 提供者年月日
- 提供種別
- 提供数
- 盛付け数
- 提供者氏名

図 3.9　メニュー提供者データベース

3.1.2.7　個人データベース

個人データベースでは，入院患者に関するデータを格納する．患者の生年月日，病気名や身長・体重などのパーソナルデータを主に扱う．図 3.10 に個人データベースの画面を示す．以下に個人データベースに格納されているデータを示す．

- 個人 ID
- 個人氏名
- フリガナ
- 生年月日
- 性別
- 病気名
- 身長
- 体重
- 職業
- 患者コード

図 3.10　個人データベース

3.1.2.8 個人摂取履歴

個人摂取履歴では, 個人データベースを基にした患者個人のデータや患者のこれまでの摂取履歴を閲覧することが可能である．図 3.11 に個人摂取履歴閲覧画面を示す．以下に閲覧できる内容を示す．

- **個人 ID**
- **個人氏名**
- **フリガナ**
- **生年月日**
- **性別**
- **RFID チップ（トレイ判別時に使用）**
- **病名**
- **身長**
- **体重**
- **標準体重**
- **BMI 値**
- **患者コード**

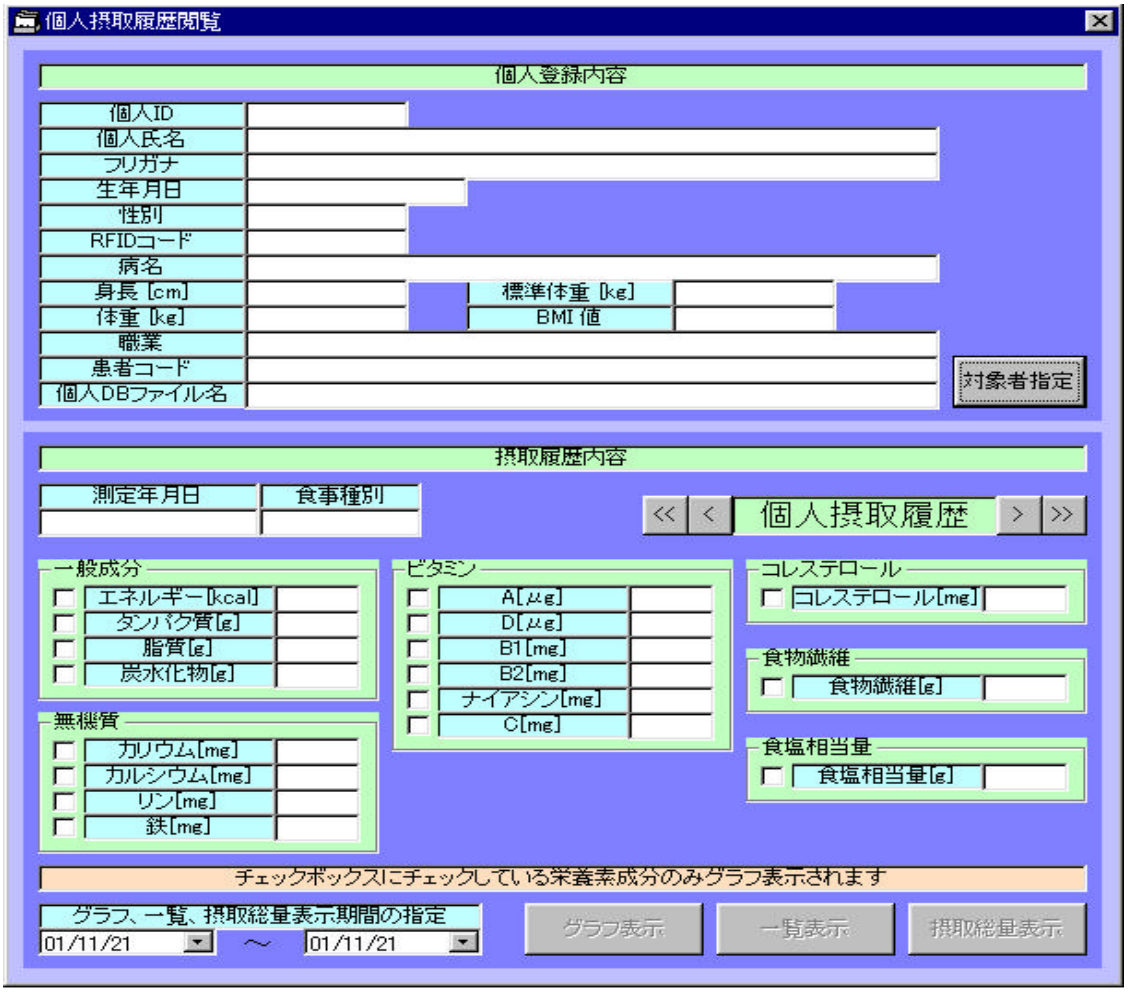


図 3.11　個人摂取履歴

3.2　動作フロー

本節では，現在，開発を進めている食事摂取量計測システムの動作フローを示す．図 3.12 に本システムの動作フロー図を示す．動作の流れとしては，まず，カメラにより食膳を撮影し食膳画像の取得を行う．次に，その食膳画像から食器部分を抽出する．そして，その食器画像から食材部分を抽出し，その面積を測定する．最後に，測定された面積量から摂取カロリー数を算出し，データベース部に登録する．本研究では，図 3.12 に示す動作フローより，特に食器抽出機構の開発を行う．

3.2　動作フロー

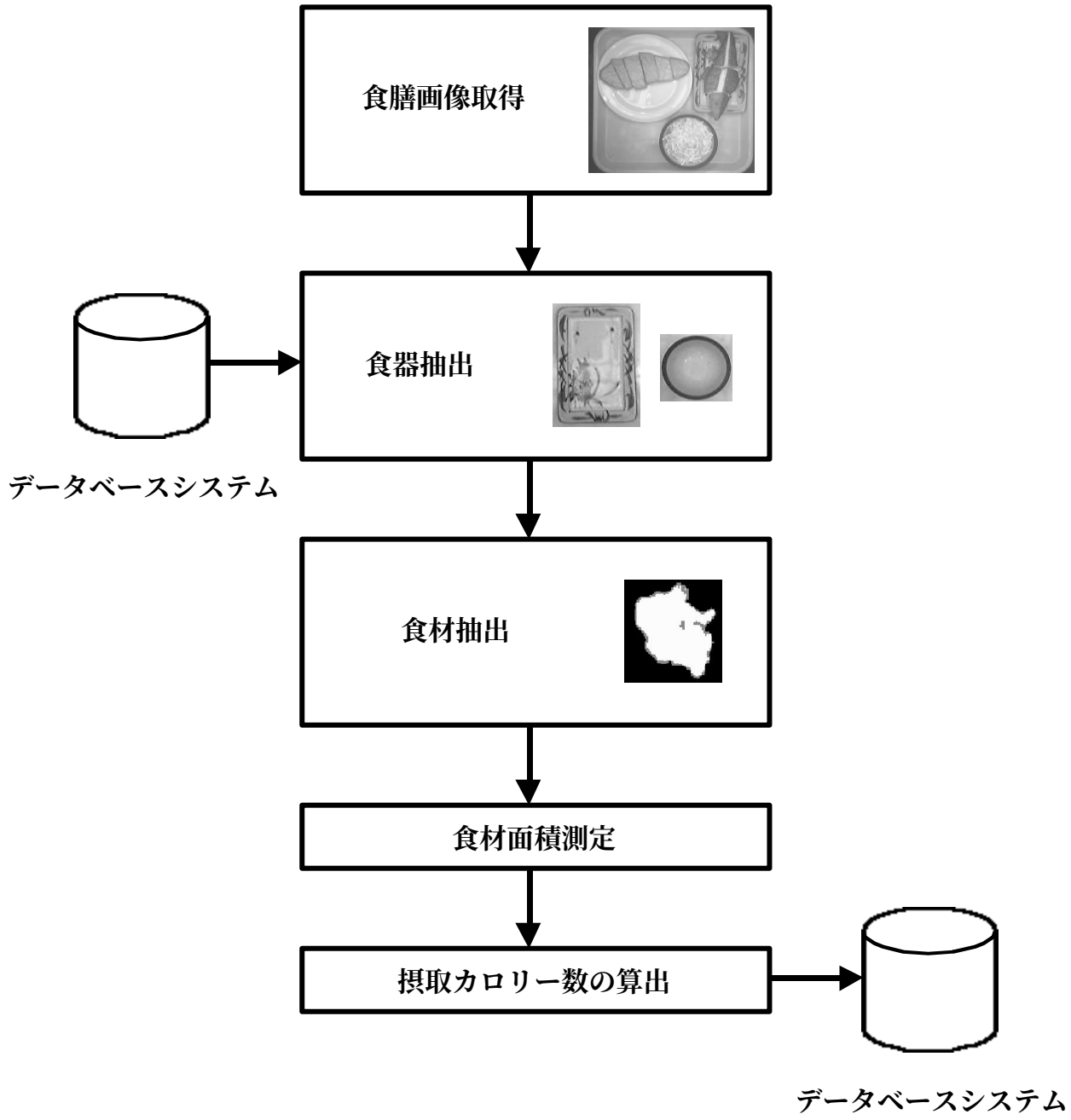


図 3.12　動作フロー図

第 4 章　NN による食器抽出処理

4.1　食器抽出手法

これまでの研究[1]-[3]では，トレイからの食器抽出が考慮されていなかった．そこで，本研究では，図 4.1 に示すような NN を用いた食器抽出処理の開発を行う[4]-[6]．食器抽出処理に NN を用いる理由[7]-[26]として，通常の画像処理[27]-[29]による全画素探索に比べ，画像探索時間が短縮可能である．また，食器から食材がはみ出してしまっているような曖昧な画像にも対応できる点がある．

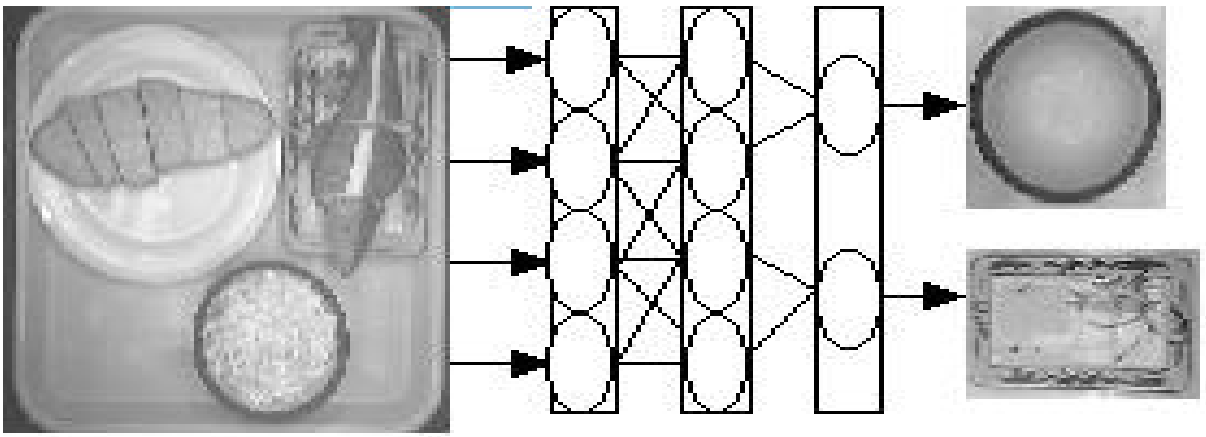


図 4.1　NN による食器抽出処理

4.2　動作フロー

本節では食事摂取量計測システムにおける食器抽出処理部分の動作フローを示す．食器抽出処理の流れは，まず，PC に接続されている USB カメラによって食膳を撮影し，食膳画像の取得を行う．次に，取得された食膳画像の 2 値化を行う．食膳画像の 2 値化の際に必要な情報はデータベースシステムとの通信によって取得される．そして，食膳画像から食器大の画像を次々に切り出していき NN に入力する．最後に，NN による判定を行い食器画像の取得を行う．図 4.2 に動作フロー図を示す．

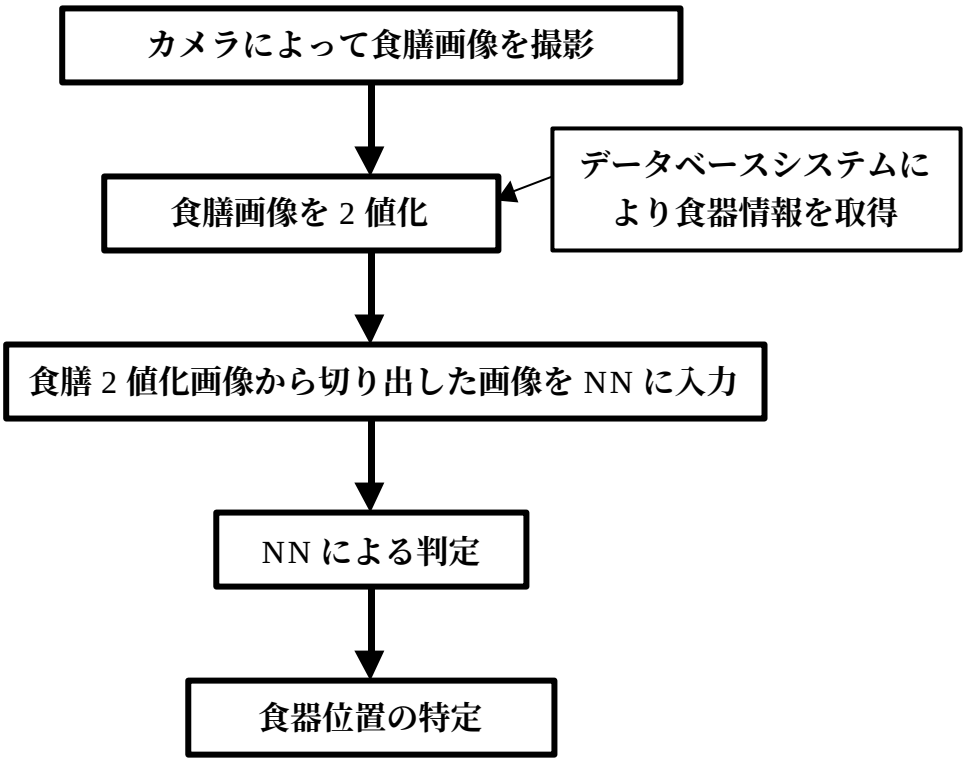


図 4.2　動作フロー図

4.2.1　画像の 2 値化

USB カメラによって撮影した食膳画像の 2 値化を行う．この際，濃淡値を基にしたトレイ上の食器に最も適したしきい値を設定し，このしきい値を用いて 2 値化を行う．図 4.3 に 2 値化画像の一例を示す．この画像はサークルで示された食器に最も適したしきい値を用いて 2 値化を行った例である．

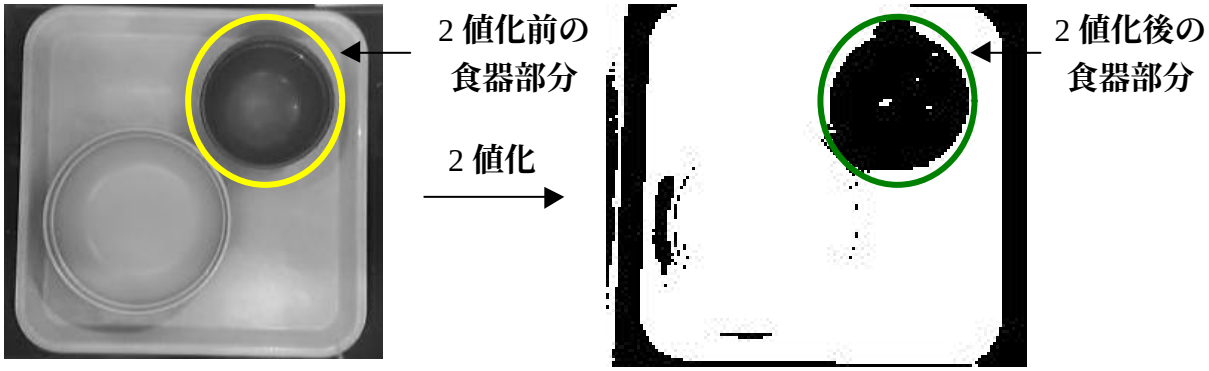


図 4.3　2 値化画像

＜食器情報の取得＞

画像の 2 値化の際に必要な食器の情報は，処理部分と食器データベースシステムとの間でファイルを介した通信によって取得する．この際の食器の情報とは，トレイに載っている食器の数，食器の種類，形状，2 値化のしきい値などを示している．図 4.4 にデータベースシステムの動作画面の一部を示す．

図 4.4　データベースシステム

4.2.2　NN による判定

次に，4.2.1 で示した方法で 2 値化を行った食膳画像の切り出しを行い NN に入力する．食膳画像の切り出し方法として，2 値化された食膳画像から一定の間隔で食器大の画像を切り出して，その切り出し画像から抽出された形状を NN に入力する．この際，使用する一定の間隔というのは現在 10 ピクセルに設定する．図 4.5 に画像を切り出し NN に入力する様子を図解したものを示す．

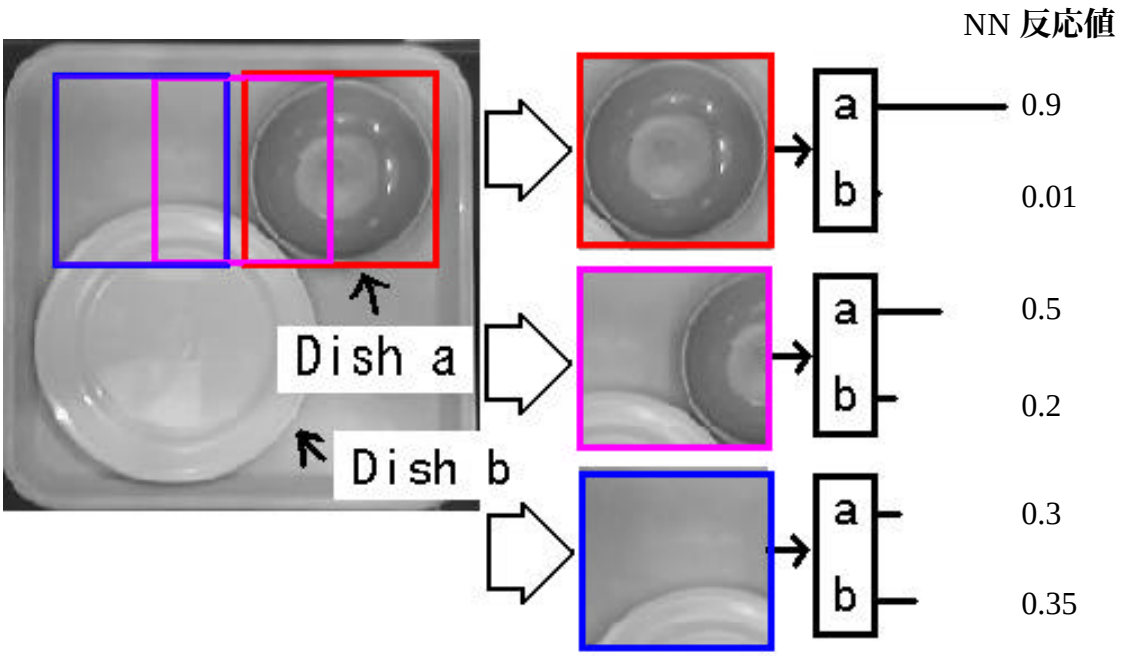


図 4.5　切り出し画像の入力

NN に入力する食器の半径・縦・横の長さ（ピクセル数）は以下のようなアルゴリズムによって求める．

(1)　2 値化のためのしきい値を設定する．しきい値より大きいか小さいかで，”1”画素と”0”画素に分ける．

(2)　8 近傍収縮処理を使用し，2 値化画像のノイズ除去を行う．

(3)　“1”画素が集合している（8 近傍に”1”画素がある）部分にそれぞれラベル番号をふる．

(4)　ラベル領域の x 軸，y 軸のそれぞれ最大値，最小値を求める．

(5)　求めた最大値，最小値から食器の縦，横の長さを求める．

切り出し画像を NN に入力した後，NN 反応が食器データベースシステムより取得した情報を NN に入力した時の反応と同じになった場合，その切り出し画像が目的の食器を含むものと判断される．それにより食器位置を特定し，抽出を行う．

4.3　食器抽出用 NN の構造

食器抽出用 NN の構造は，中間層の数を 1 に固定した 3 層の階層型 NN である．入力層細胞数は 50，中間層細胞数は 50，出力層細胞数は 10 である．この NN の入力層には，食器の外縁を取り出す際に使用するしきい値，食膳画像における食器の半径・縦・横の長さ（ピクセル数），食器の形状，食器の濃淡画像の特徴値が入力値として入力される．また，出力ユニット数は登録した食器数に対応する．NN によって出力された値のうち，最も高い値を出力したパターンに相当する食器が，入力された食器と特定され，抽出される．図 4.6 に食器抽出用 NN の構造を示す．

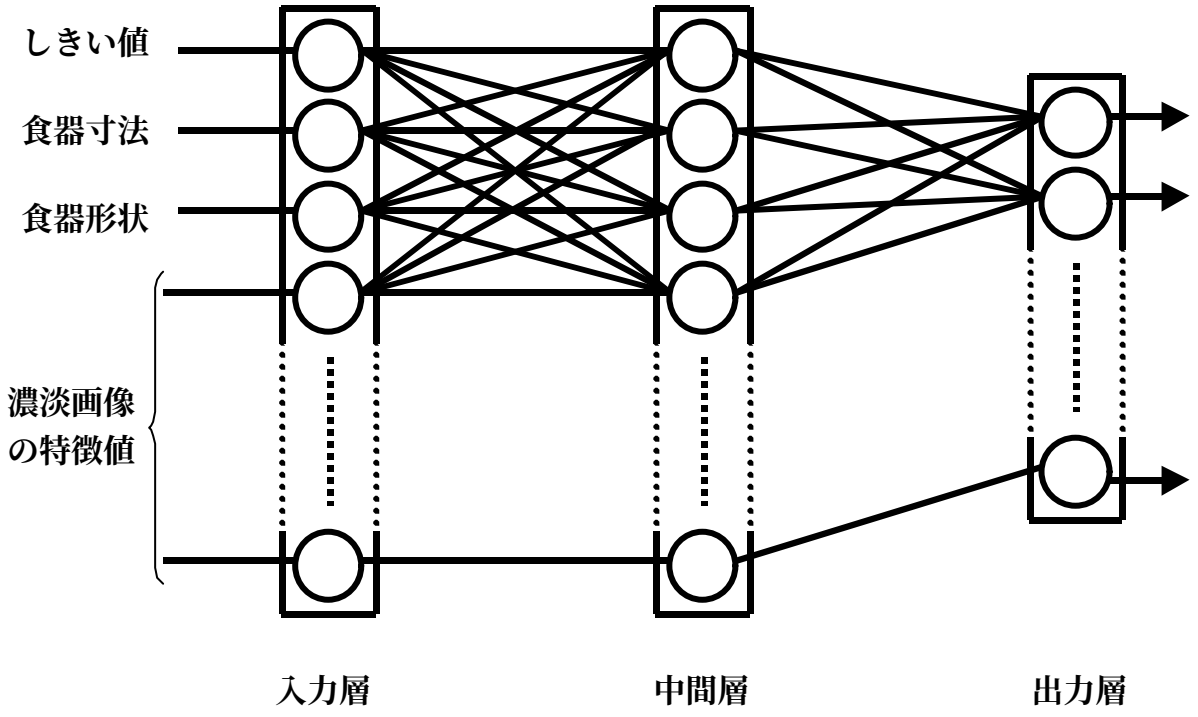


図 4.6　食器抽出用 NN の構造

第 5 章　食器抽出実験

本章では食事摂取量計測システムを用いて行った 2 つの食器抽出実験について述べる．実験の目的は，4.で述べたような食器抽出機構の機能確認のためである．

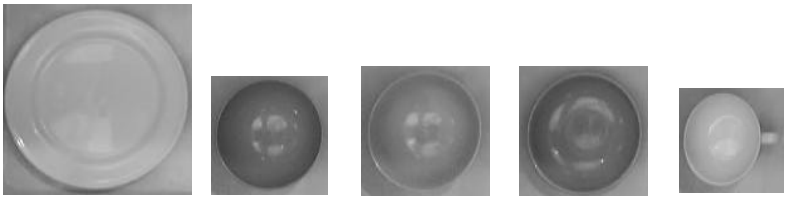
5.1　食器抽出実験 1

残食材量 0%の食器画像（食材の載っていない食器画像）を使用しての食器抽出実験を行った．

5.1.1　実験条件

本実験の実験条件を以下に示す．使用する食器は実際病院内で使用される食器である．使用する食器の数は図 5.1 と表 5.2 に示すとおり 10 種類の食器を使用する．また，図 5.2 と表 5.3，表 5.4 に示すとおり 10 種類の食器使用し受給者 3 人に対して 59 種類のメニュー（2002 年 1 月 1 日朝食～2002 年 1 月 20 日昼食）を用意する．NN の学習・評価条件は表 5.1 に示す．また，本実験では内縁範囲指定値というパラメータの値を 0%～100%のうち 5%区切りで設定する．この内縁範囲指定値とは食器内での食材の占める割合を事前に予測する数値である．実験に使用するデータ数は 59（メニュー種類数）×21（パラメータ値の数）である．

食器 ID：1　食器 ID：2　食器 ID：3　食器 ID：4　食器 ID：5



食器 ID：6　食器 ID：7　食器 ID：8　食器 ID：9　食器 ID：10

図 5.1　使用食器画像

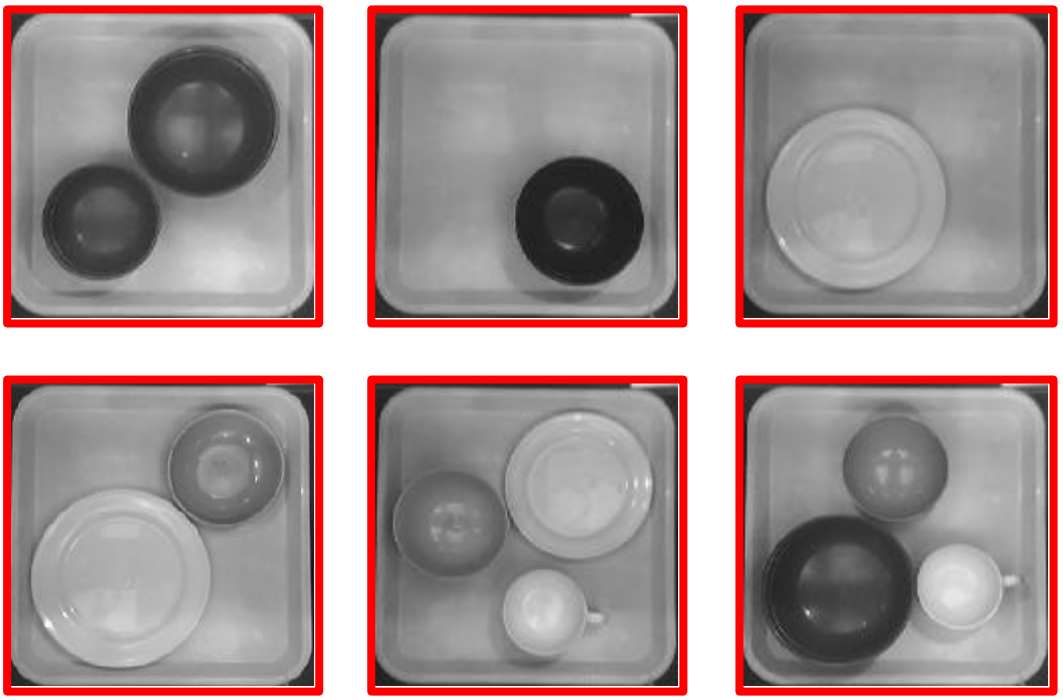


図 5.2　食器配膳例

表 5.1　NN の学習および評価条件

評価パターン数（食器の種類数）	10
各パターンの学習枚数	5
各パターンの評価枚数	20
2 乗平均誤差	1.0E－4
学習方法	改良型 BP 法

表 5.2　食器情報

食器 ID	食器名	形状コード	横寸法（画素）	縦寸法（画素）	スレッシュホールド値
1	プラスティック碗 1	0	31	31	130
2	プラスティック碗 2	0	24	24	105
3	プラスティック碗 3	0	31	31	105
4	プラスティック碗 4	0	27	27	105
5	皿 1	0	30	30	157
6	皿 2	0	36	36	157
7	茶碗 1	0	24	24	120
8	茶碗 2	0	23	23	135
9	小皿	0	24	24	125
10	カップ	0	18	18	150

表 5.3 メニュー一覧（1）

メニューID	食器										提供日	食事種別	受給者		
	1	2	3	4	5	6	7	8	9	10			1	2	3
1		○				○					2002/01/01	朝	○	○	
2						○	○				2002/01/01	昼	○		
3						○			○		2002/01/01	夕	○	○	○
4						○		○			2002/01/02	朝	○		
5						○				○	2002/01/02	昼	○		
6		○				○				○	2002/01/02	夕	○		
7						○	○			○	2002/01/03	朝	○		
8						○			○	○	2002/01/03	昼	○	○	
9						○		○		○	2002/01/03	夕	○	○	○
10	○	○									2002/01/04	朝	○	○	
11	○						○				2002/01/04	昼	○	○	○
12	○								○		2002/01/04	夕	○	○	○
13	○							○			2002/01/05	朝	○	○	
14		○	○								2002/01/05	昼	○	○	
15			○				○				2002/01/05	夕	○	○	○
16			○						○		2002/01/06	朝	○		
17			○					○			2002/01/06	昼	○		
18		○			○						2002/01/06	夕	○	○	○
19					○		○				2002/01/07	朝	○	○	○
20					○				○		2002/01/07	昼	○		
21					○			○			2002/01/07	夕	○	○	○
22		○		○							2002/01/08	朝	○		
23				○			○				2002/01/08	昼	○	○	
24				○					○		2002/01/08	夕	○		
25				○				○			2002/01/09	朝	○	○	
26	○									○	2002/01/09	昼	○	○	○
27			○							○	2002/01/09	夕	○	○	○
28					○					○	2002/01/10	朝	○		
29				○						○	2002/01/10	昼	○	○	
30	○	○								○	2002/01/10	夕	○	○	○

表 5.4 メニュー一覧（2）

メニュー ID	食器										提供日	食事種別	受給者		
	1	2	3	4	5	6	7	8	9	10			1	2	3
31	○						○			○	2002/01/11	朝	○		
32	○								○	○	2002/01/11	昼	○	○	
33	○							○		○	2002/01/11	夕	○	○	
34		○	○							○	2002/01/12	朝	○		
35			○				○			○	2002/01/12	昼	○	○	○
36			○						○	○	2002/01/12	夕	○	○	○
37			○					○		○	2002/01/13	朝	○		
38		○			○					○	2002/01/13	昼	○		
39					○		○			○	2002/01/13	夕	○		
40					○				○	○	2002/01/14	朝	○		
41					○			○		○	2002/01/14	昼	○	○	
42		○		○						○	2002/01/14	夕	○	○	○
43				○			○			○	2002/01/15	朝	○	○	○
44				○					○	○	2002/01/15	昼	○	○	○
45				○				○		○	2002/01/15	夕	○	○	○
46		○								○	2002/01/16	朝	○	○	○
47							○			○	2002/01/16	昼	○	○	○
48									○	○	2002/01/16	夕	○	○	○
49								○		○	2002/01/17	朝	○	○	
50	○										2002/01/17	昼	○	○	○
51		○									2002/01/17	夕	○	○	○
52			○								2002/01/18	朝	○	○	○
53				○							2002/01/18	昼	○	○	○
54					○						2002/01/18	夕	○	○	○
55						○					2002/01/19	朝	○	○	○
56							○				2002/01/19	昼	○	○	○
57								○			2002/01/19	夕	○	○	○
58									○		2002/01/20	朝	○	○	○
59										○	2002/01/20	昼	○	○	○

5.1.2　実験結果

実験結果としてすべての食器の抽出が行えた．食器抽出処理画面を図 5.3 と図 5.4 に示す．図 5.3 は正確な位置での食器抽出の例を示す．図 5.4 は正確な位置で食器抽出が行えなかった例を示す．食器抽出処理の実験結果を表 5.5〜表 5.7 に示す．表の行にはパラメータ値，列にはメニューID を示す．表中の○はトレイ上のすべての食器が正確な位置で確定できたものを表す．△は一部の食器のみ正確な位置で確定できたものを表す．△の横の数値は正確な位置で食器抽出が行えた食器 ID を表す．×は正確な位置で食器抽出が行えなかったものを表す．実験結果より，図 5.4 に示すように食器位置を確定させるためのサークルが食器からずれた位置で食器位置と確定させる場合が存在した．この原因として，食膳画像の 2 値化を行う際に正確なしきい値の設定が困難なため，NN への入力画像にノイズ等が発生したためと考察する．

現在，NN への入力画像は濃淡画像を使用し，濃淡値によるしきい値を用いて食膳画像の 2 値化を行っている．しかしながら，濃淡値による 2 値化処理では光の反射などの影響でしきい値の設定が難しい．そのため，2 値化画像を作るための画像を濃淡画像からカラー画像に変更する．さらに，2 値化のためのしきい値設定においてもこれまでの濃淡値によるしきい値設定から明度・色相値によるしきい値設定に変更することを検討する．この変更により，食膳画像の 2 値化画像が鮮明になると予想される．

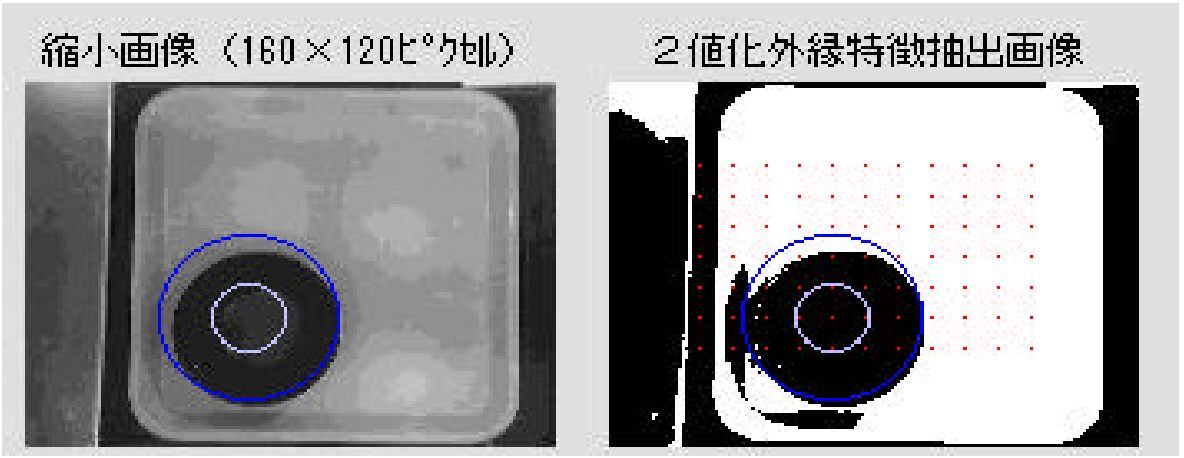


図 5.3　食器位置確定成功例

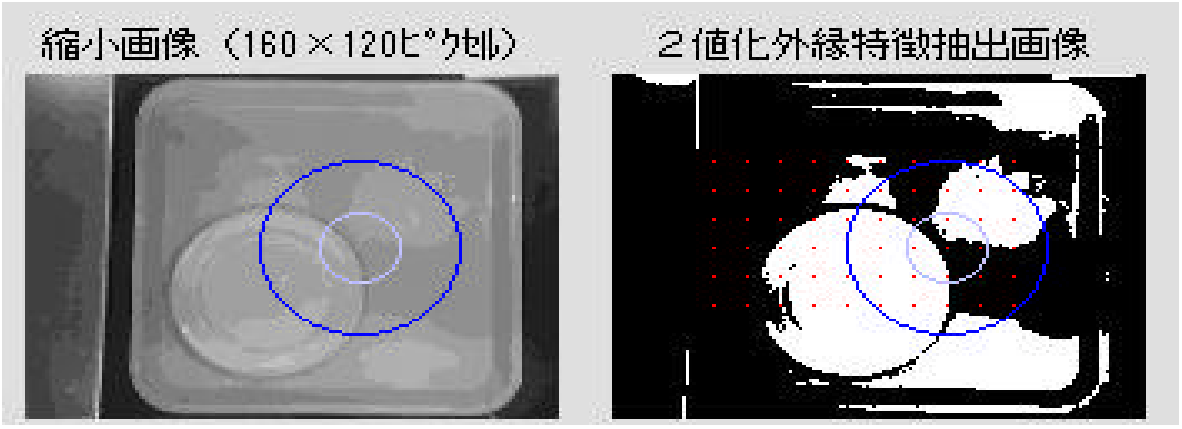


図 5.4　食器位置確定失敗例

表 5.5 **実験結果**（1）

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
5%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
10%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
15%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
20%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
25%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
30%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
35%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
40%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
45%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
50%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
55%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
60%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
65%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
70%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
75%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
80%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
85%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
90%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
95%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
100%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×

表 5.6　実験結果（2）

	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
0%	×	△4	△4	△4	△4	△2	×	×	×	×	×	×	○	×	×	×	×	×	×
5%	×	△4	△4	△4	△4	△2	×	×	×	×	×	×	○	×	×	×	×	×	×
10%	×	△4	△4	△4	△4	△2	△7	×	×	×	×	×	○	×	×	○	×	×	×
15%	×	△4	△4	△4	△4	△2	△7	×	×	×	×	×	○	×	×	○	×	○	×
20%	×	△4	△4	△4	△4	△2	△7	×	×	×	×	×	○	×	×	○	×	○	×
25%	×	△4	△4	△4	△4	△2	△7	×	×	×	×	×	○	×	×	○	×	×	×
30%	×	△4	△4	×	△4	△2	△7	×	×	×	×	×	○	×	×	○	×	×	×
35%	×	△4	×	×	△4	△2	△7	×	×	×	×	×	○	×	×	○	×	×	×
40%	×	△4	×	×	×	△2	△7	×	×	×	×	×	○	×	×	○	×	×	×
45%	×	△4	×	×	×	△2	×	×	×	×	○	×	×	×	×	○	×	×	×
50%	×	△4	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
55%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
60%	×	△4	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
65%	×	△4	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
70%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
75%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
80%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
85%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
90%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
95%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
100%	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×

5.2　食器抽出実験 2

この実験では，炊飯米を載せた食器画像を用いての食器抽出実験を行った．

5.2.1　実験条件

5.1 の実験と同様，食事摂取量計測システムを用いて，食器上に食材が載っている場合の食器位置確定実験を行う．使用する食器は，5.1 での食器抽出実験において比較的正確な位置で食器抽出が行えた食器（食器 ID：4，食器 ID：7）を使用する．計測対象食材は炊飯米とする．炊飯米の残食パターンは 0%，20%，40%，60%，80%，100%の 6 パターンとする．食器の配置パターンは図 5.5 に示す 5 パターンとする．また，食器位置特定実験の際に使用した内縁範囲指定値は 0%に設定する．この理由は食器全体を探索し，食器上の食材が本実験の結果にどう影響を及ぼすかを調べるためである実験使用データ数は 2（使用食器数）×6（残食パターン数）×5（食器配置パターン数）の 60 である．図 5.6～図 5.17 に実験使用画像を示す．図 5.6～図 5.11 には食器 ID：4 を使用した画像データを示す．図 5.12～図 5.17 には食器 ID：7 を使用した画像データを示す．

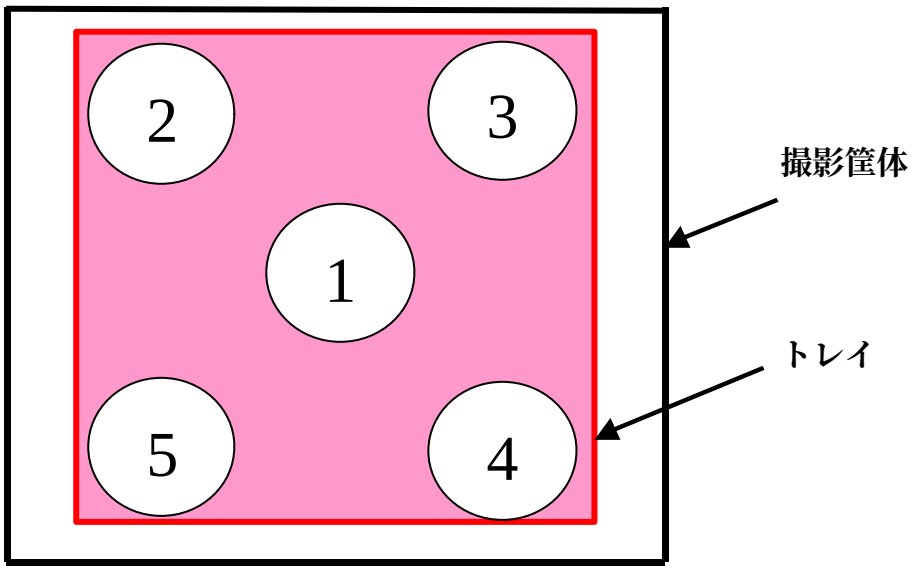


図 5.5　食器の配置

図 5.6　実験データ（1）：残量 0%

図 5.7　実験データ（1）：残量 20%

図 5.8　実験データ（1）：残量 40%

図 5.9　実験データ（1）：残量 60%

図 5.10　**実験データ（1）：残量** 80%

図 5.11　**実験データ（1）：残量** 100%

図 5.12　**実験データ（2）：残量** 0%

図 5.13　**実験データ（2）：残量** 20%

図 5.14　**実験データ（2）：残量** 40%

図 5.15　**実験データ（2）：残量** 60%

図 5.16　**実験データ（2）：残量** 80%

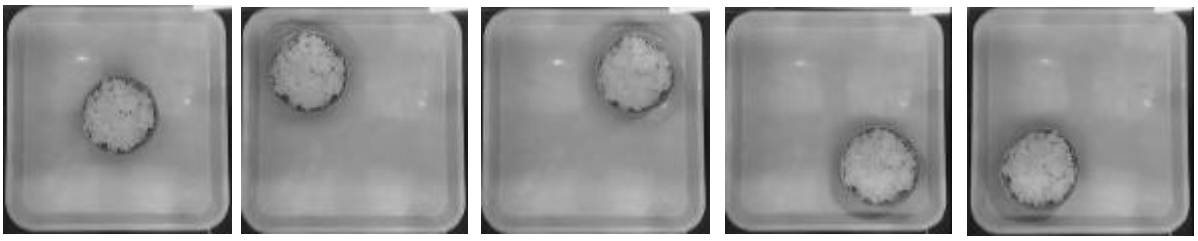


図 5.17　**実験データ（2）：残量** 100%

5.2.2　実験結果

本実験の実験結果を表 5.7 と表 5.8 に示す．表 5.7 には食器 ID：4 を使用した実験結果を示す．表 5.8 には食器 ID：7 を使用した実験結果を示す．表中の○は正確な位置で食器抽出が行えたものを示す．×は正確な位置で食器抽出が行えなかったものを示す．図 5.18 に抽出成功時のシステム動作画面，図 5.19 に抽出失敗時のシステム動作画面を示す．

表 5.7　実験結果（1）

食器 ID 4	残量 0%	配置1	○
		配置2	○
		配置3	○
		配置4	○
		配置5	○
	残量 20%	配置1	×
		配置2	×
		配置3	×
		配置4	×
		配置5	×
	残量 40%	配置1	×
		配置2	×
		配置3	×
		配置4	×
		配置5	×
	残量 60%	配置1	×
		配置2	×
		配置3	×
		配置4	×
		配置5	×
	残量 80%	配置1	×
		配置2	×
		配置3	×
		配置4	×
		配置5	×
	残量 100%	配置1	×
		配置2	×
		配置3	×
		配置4	×
		配置5	×

表 5.8　実験結果（2）

食器 ID 7	残量 0%	配置1	○
		配置2	×
		配置3	×
		配置4	×
		配置5	×
	残量 20%	配置1	×
		配置2	×
		配置3	×
		配置4	×
		配置5	×
	残量 40%	配置1	×
		配置2	×
		配置3	×
		配置4	×
		配置5	×
	残量 60%	配置1	×
		配置2	×
		配置3	×
		配置4	×
		配置5	×
	残量 80%	配置1	×
		配置2	×
		配置3	×
		配置4	×
		配置5	×
	残量 100%	配置1	×
		配置2	×
		配置3	×
		配置4	×
		配置5	×

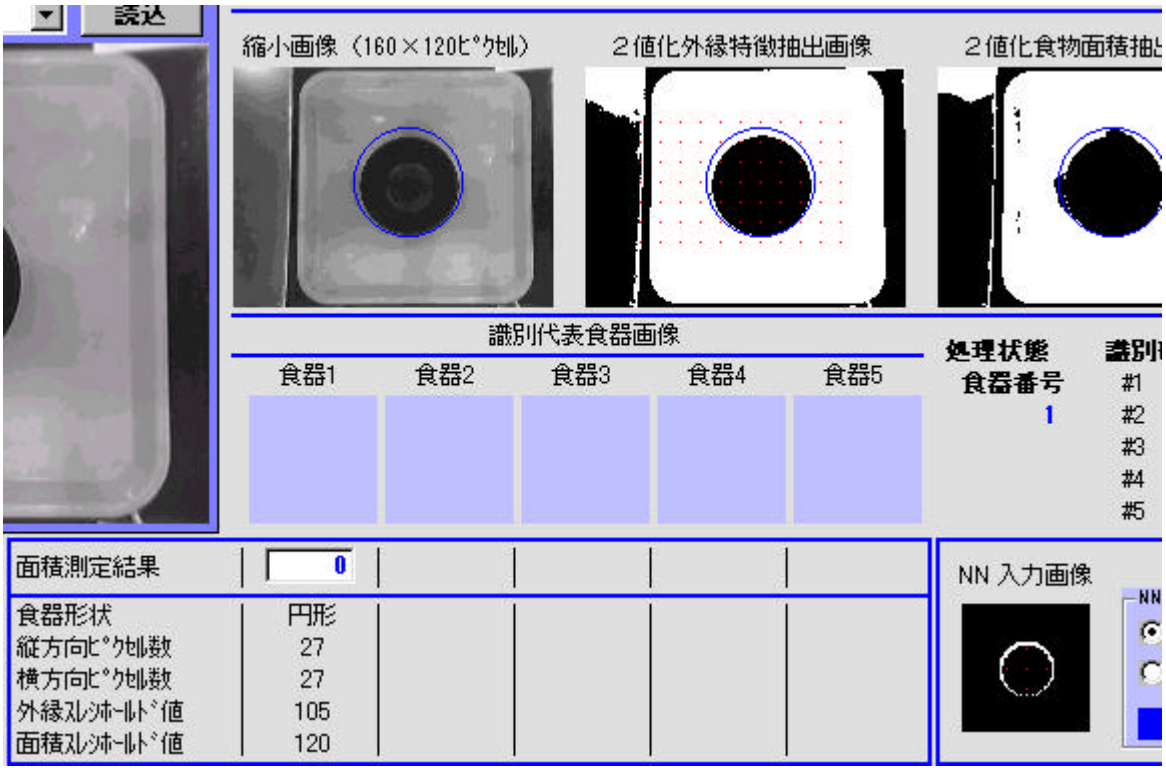


図 5.18　抽出成功例

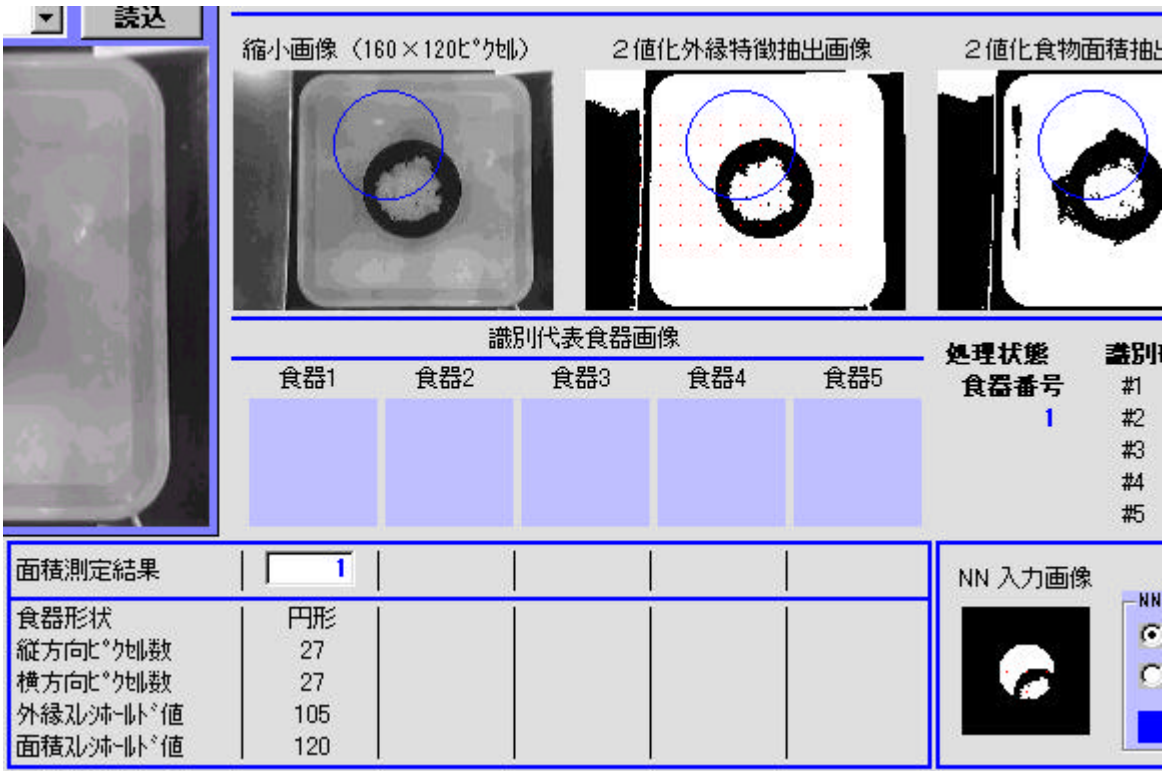


図 5.19　抽出失敗例

実験結果より，食器 ID：4 で残食 0%の 5 パターン，食器 ID：7 で残食 0%のうちの 1 パターン，計 6 パターンのみ抽出成功が確認できた．これより，食材が載っている画像は全て食器抽出失敗したということになる．これは，食材部分がノイズとなってしまうため，NN に入力しても食器として認識しないためである．

5.3　考察

本章では，食事摂取量計測システムにおける食器抽出機構の性能確認実験を行った．食材が無い場合の食器抽出実験（実験 1）においては，全ての食器抽出に成功した．しかしながら，正確な食器位置特定が行えなかったパターンが複数存在した．その理由として，NN 入力画像の元となる 2 値化画像が不鮮明であるということである．この問題の解決策として，現在モノクロ画像を濃淡値によるしきい値によって 2 値化を行っているが，それをカラー画像を明度，色相値によるしきい値によって 2 値化を行う方式に代えるということが挙げられる．食器上に食材が載った場合での食器抽出実験においては，ほとんどの食器位置特定を行うことができなかった．その理由として，食器抽出用 NN の学習には食材の載っていない食器のみの画像を使用する．そのため，食材が載っている食器画像では，NN に食器として判定されないためと考察される．

第 6 章　食物除去アルゴリズム

5.3 で示した，食物が載った食器を用いての食器抽出実験結果より，食器位置特定成功率の低い結果となった．その理由として考えられることに，食膳画像の 2 値化の際，食物部分がノイズとなってしまうということが挙げられる．そのため，2 値化画像から画像を切り出して正しい画像を NN に入力するため，切り出し画像の食物部分を除去する必要がある．そこで，本章では食物除去アルゴリズムを考案する．

6.1　アルゴリズムの概要

4.2.3 で示したように，食事摂取量計測システムの食器抽出機構には，2 値化された食膳画像から小さな画像を切りだし NN に入力する処理部分が存在する．今回考案するアルゴリズムは，NN への入力用画像を作成する際に食物が含まれている場合，食物部分を除去する．図 6.1 に正しい入力画像を，図 6.2 に食物がノイズとなってしまっている入力画像を示す．

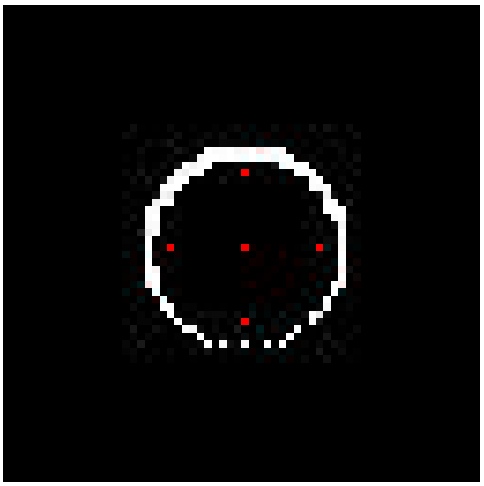


図 6.1　NN 入力画像（1）

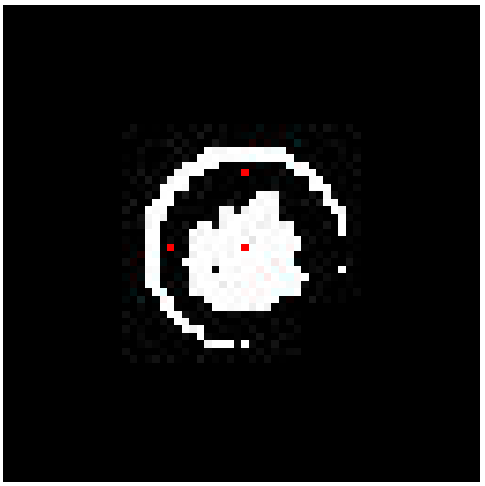


図 6.2　NN 入力画像（2）

食物除去の方法は，NN 入力画像の元となる食膳 2 値化画像の全画素を探索する際に，食物があると推測される X 座標においての食物部分の Y 座標の最小値，最大値を検出し最小値から最大値にあたる部分を黒画素で埋める．詳細に述べると，切り出し画像を上から下に走査していき食物部分の上端の座標，下端の座標を検出し，その座標の間を黒画素で埋めるという作業を全ての X 座標において行いながら，NN 入力画像を作る．図 6.3 に食物除去アルゴリズムを図解したものを示す．

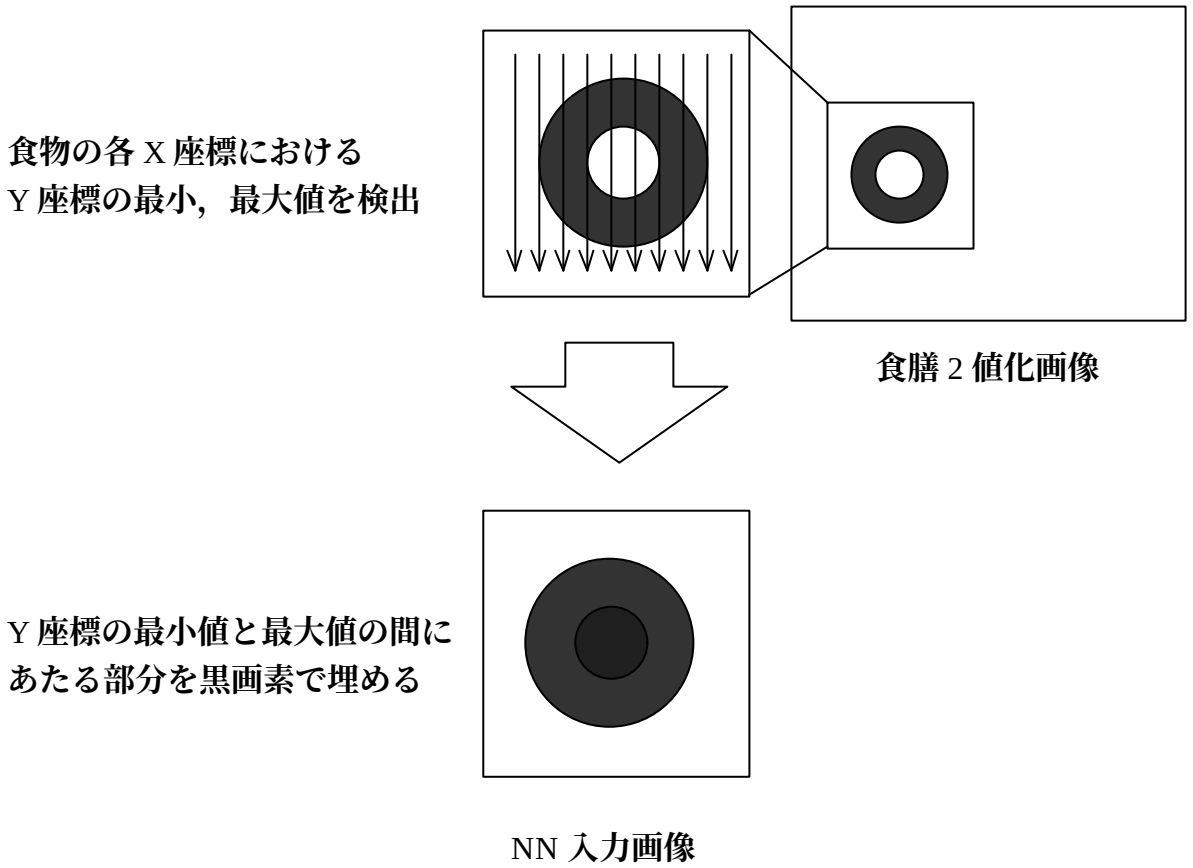


図 6.3　食物除去アルゴリズム

6.2　動作フロー

食物除去処理を示す．このとき，食器部分は黒（0 画素），トレイと食物部分は白（１画素）である．図 6.4 に動作フロー図を示す．

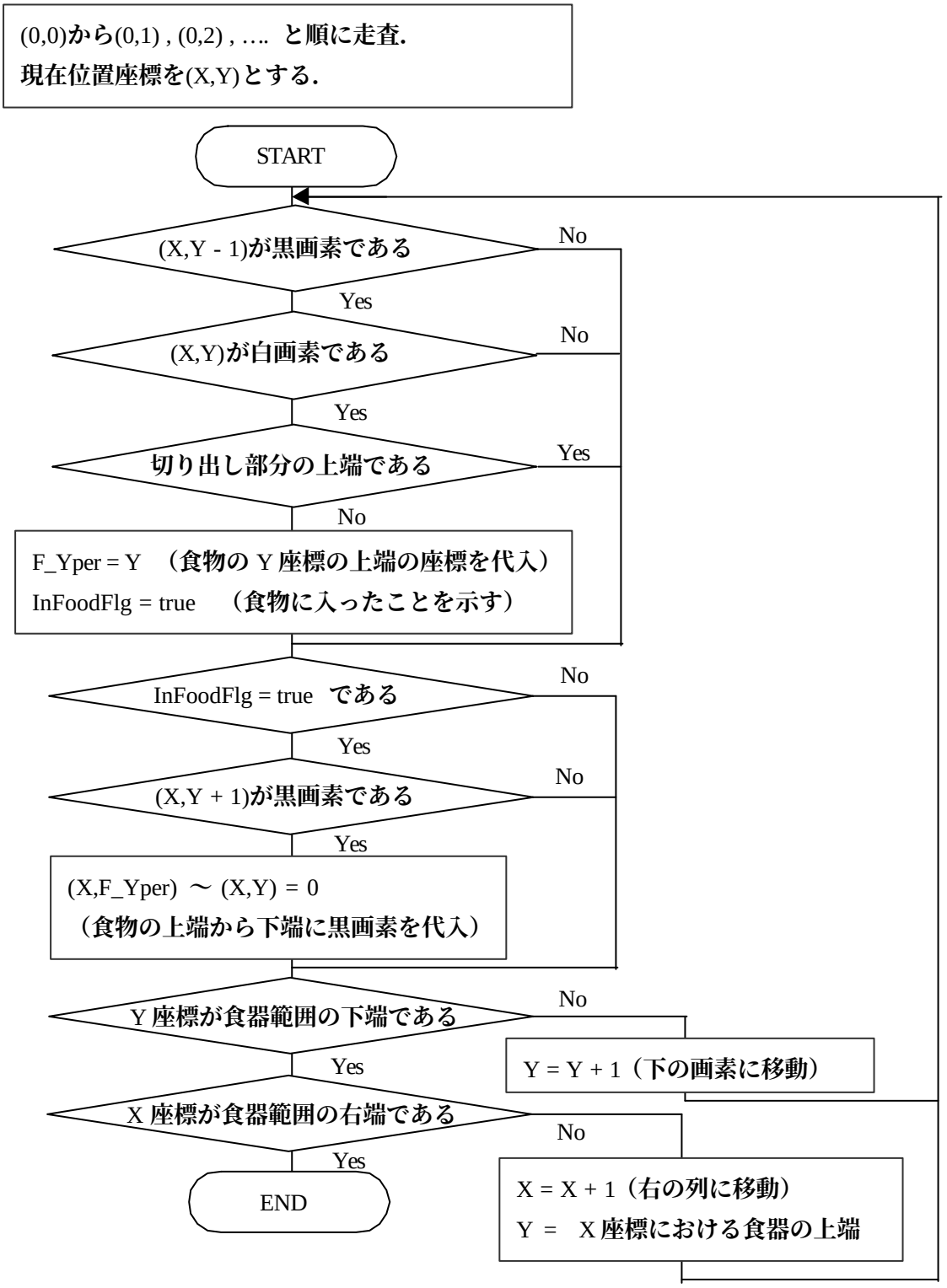


図 6.2　フローチャート

第 7 章　食物除去アルゴリズムの性能確認実験

6.で示した食物除去アルゴリズムを基に，現在の食事摂取量計測システムのプログラムに改良を加えた．本章では，その改良後のシステムを用いての食器抽出実験を行った．

7.1　実験条件

プログラム変更後の食事摂取量計測システムを用いて，食器上に食材が載っている場合の食器位置確定実験を行った．使用する食器は 5.3 で使用したものと同様の食器（食器 ID：4，食器 ID：7）を使用する．計測対象食材は炊飯米とする．炊飯米の残食パターンは 0%，20%，40%，60%，80%，100%の 6 パターンとする．食器の配置パターンは図 5.5 に示す 5 パターンとする．また，食器位置特定実験の際に使用した内縁範囲指定値は 0%に設定する．実験使用データ数は 2（使用食器数）×6（残食パターン数）×5（食器配置パターン数）の 60 である．

7.2　実験結果

本実験の実験結果を表 7.1 と表 7.2 に示す．表 7.1 には食器 ID：4 を使用した実験結果を示す．表 7.2 には食器 ID：7 を使用した実験結果を示す．表中の○は正確な位置で食器抽出が行えたものを示す．×は正確な位置で食器抽出が行えなかったものを示す．表 7.3 と表 7.4 に食器位置特定率を示す．表 7.3 には食器 ID4，表 7.4 には食器 ID7 の食器位置特定率を示す．表 7.3，表 7.4 には性能比較のため，プログラム改良前の食事摂取量計測システムを用いて行った実験結果を示す．図 7.1 に抽出成功時のシステム動作画面，図 7.2 に抽出失敗時のシステム動作画面を示す．

表 7.1 実験結果（1）

			プログラム改良後	プログラム改良前
食器ID 4	残量0%	配置1	○	○
		配置2	×	○
		配置3	○	○
		配置4	○	○
		配置5	×	○
	残量20%	配置1	○	×
		配置2	×	×
		配置3	×	×
		配置4	○	×
		配置5	×	×
	残量40%	配置1	○	×
		配置2	○	×
		配置3	○	×
		配置4	○	×
		配置5	×	×
	残量60%	配置1	○	×
		配置2	○	×
		配置3	○	×
		配置4	○	×
		配置5	×	×
	残量80%	配置1	○	×
		配置2	○	×
		配置3	○	×
		配置4	○	×
		配置5	○	×
	残量100%	配置1	○	×
		配置2	○	×
		配置3	×	×
		配置4	○	×
		配置5	×	×

表 7.2 実験結果（2）

			プログラム改良後	プログラム改良前
食器ID 7	残量0%	配置1	○	○
		配置2	×	×
		配置3	×	×
		配置4	○	×
		配置5	×	×
	残量20%	配置1	○	×
		配置2	○	×
		配置3	×	×
		配置4	○	×
		配置5	×	×
	残量40%	配置1	○	×
		配置2	×	×
		配置3	○	×
		配置4	×	×
		配置5	×	×
	残量60%	配置1	○	×
		配置2	×	×
		配置3	×	×
		配置4	○	×
		配置5	×	×
	残量80%	配置1	○	×
		配置2	×	×
		配置3	×	×
		配置4	○	×
		配置5	×	×
	残量100%	配置1	○	×
		配置2	○	×
		配置3	○	×
		配置4	○	×
		配置5	×	×

表 7.3 食器位置特定率（1）

		プログラム改良後	プログラム改良前
食器ID 4	残食0%	60%	100%
	残食20%	40%	0%
	残食40%	80%	0%
	残食60%	80%	0%
	残食80%	100%	0%
	残食100%	60%	0%
	全体	70%	16.6%

表 7.4 食器位置特定率（2）

		プログラム改良後	プログラム改良前
食器ID 7	残食0%	40%	20%
	残食20%	60%	0%
	残食40%	40%	0%
	残食60%	40%	0%
	残食80%	40%	0%
	残食100%	80%	0%
	全体	50%	3.3%

7.2　実験結果

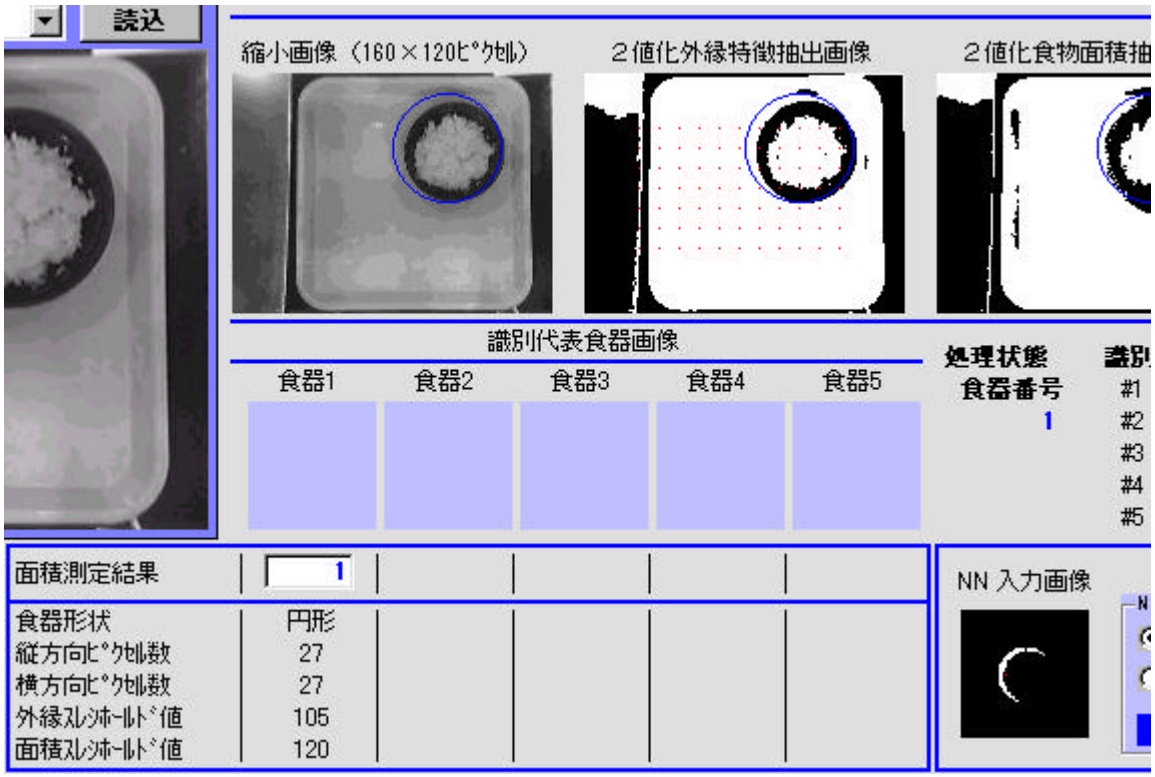


図 7.1　抽出成功例

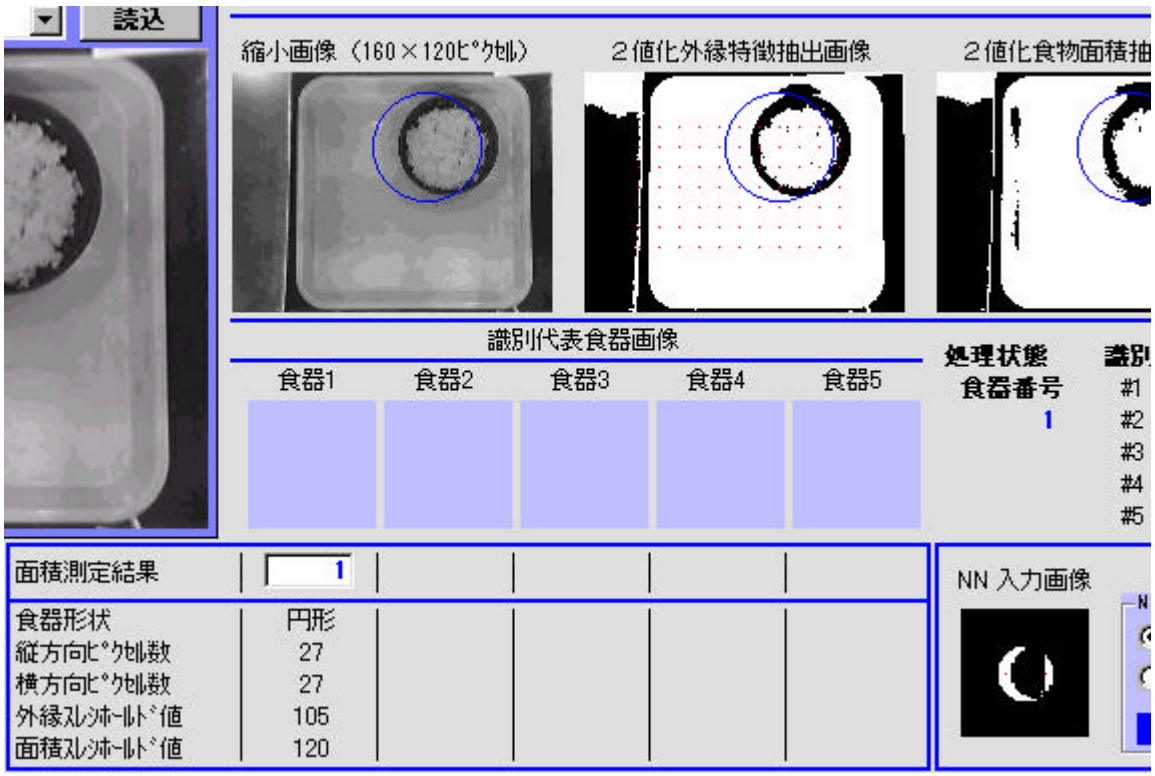


図 7.2　抽出失敗例

実験を行った結果，食器位置確定率は 60%となった．食器別に比較すると，食器 ID：4 の場合 70%，食器 ID：７の場合 50%である．プログラム改良前のシステムを用いた場合での食器位置確定率は 10%である．食器別に比較すると，食器 ID：４の場合 16.6%，食器 ID：7 の場合 3.3%である．

7.3　考察

7.2 の実験結果から食器位置確定の機能は大幅に向上したことが確認できた．図 7.3 と図 7.4 に実験結果画面の一部を示す．図 7.3 はプログラム変更後，図 7.4 はプログラム変更前のものである．図 7.3 と図 7.4 の実験条件は同じである．

図 7.3，図 7.4 内のサークルはシステムが食器位置と断定した部分を表す．図 7.3 を見て分かるとおり，食材が載っている場合でも食器位置確定に成功した．この理由として，食器抽出の際 NN に入力される切り出し画像の食物部分が除去されているからである．図 7.5 と図 7.6 に食器部分が切り出されたときの，NN へ入力する切り出し画像を示す．図 7.5 はプログラム変更後の場合，図 7.6 はプログラム変更前の場合である．

図 7.5 と図 7.6 を見比べて，プログラム変更後の NN 入力画像は，食物部分（図 7.6 に見られる，中心の白画素部分）が黒画素で埋められていることが分かる．そのため，完全な食器画像が入力された状況と同じになり食器位置確定が成功する．しかしながら，食器位置特定に失敗した例も幾つか見られた．その理由としては，アルゴリズムの性質上，2 値化画像のトレイにあたる部分にノイズとして黒画素があった場合，ノイズとノイズの間，もしくは，ノイズと食器の間も黒画素で埋めてしまうからだと考察される．この問題の解決策として，食膳画像の 2 値化の精度を高め，鮮明な 2 値化画像を作成することが必要である．その他の問題としては，プログラムを改良することで，処理時間が遅くなったことが挙げられる．今後はこれらの問題を改善するために，プログラムに再度改良を加えることが課題となる．

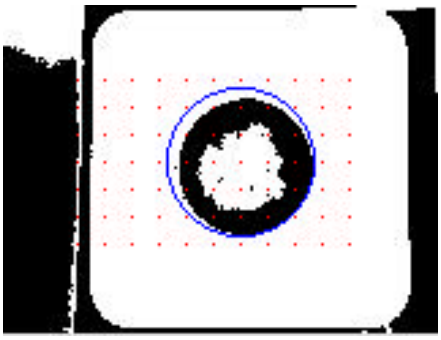


図 7.3 実験結果（プログラム変更後）

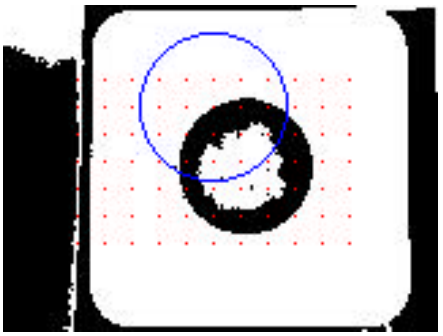


図 7.4 実験結果（プログラム変更前）

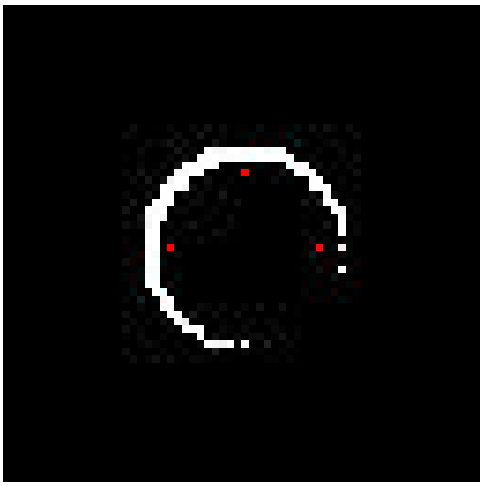


図 7.5 NN 入力画像（プログラム変更後）

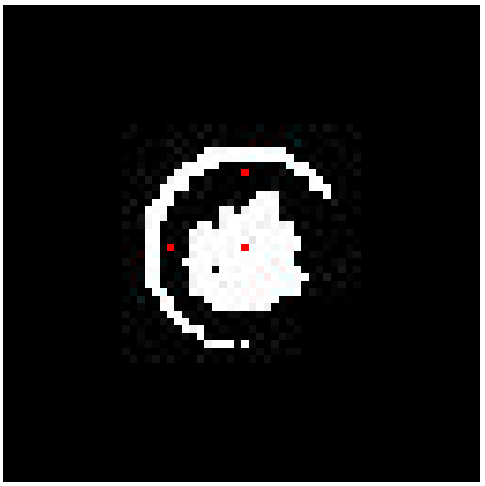


図 7.6 NN 入力画像（プログラム変更前）

第 8 章　まとめ

本研究では，食事摂取量計測システムにおける食器抽出処理の開発を行った．食器抽出機構の性能を確認するため，病院内で使用される食器を用いた実験，及び，炊飯米の載った食器を用いた実験を行った．実験を行った結果，正しい食器位置を確定できないパターンが複数見られた．その理由として，食膳画像の 2 値化画像が不鮮明であることや，食物が載っている場合，食物部分がノイズとなってしまうため，NN に正しい画像が入力されないことなどが挙げられる．そこで，NN への入力画像の食物部分を一時的に除去するアルゴリズム，食物除去アルゴリズムの提案を行った．

食物除去アルゴリズムを基に食事摂取量計測システムのプログラムに改良を加え，食物が載った食器にも対応出来るシステムの開発を行った．食物除去アルゴリズムの有効性を示すため，改良を加えたシステムを用いて，炊飯米の載った食器の食器抽出実験を行った．実験を行った結果，食器位置特定率は 60%となった．プログラム改良前のシステムを用いて同じ実験を行った場合の食器位置特定率が 10%であった. この結果より大幅に機能向上が図れたと言える．

改良後のシステムは食物アルゴリズムの性質上，食器以外の位置に何らかのノイズによって黒画素がある場合うまく動作しない場合がある．今後は，このノイズを無くす 2 値化処理の方法の再検討や，ノイズに対応出来るアルゴリズムの提案を課題とする．

第 9 章　謝辞

2 年間に渡りご指導していただいた高知工科大学情報システム工学科の竹田史章教授に心から感謝いたします．また，本システムの研究開発に協力していただいたグローリー工業（株）の吉田邦夫様，グローリー商事（株）の長田司郎様，ヴィスト（有）の小林茂樹社長殿，形部麗子様，エヌビーサイト（有）の尾江俊昭社長殿，高知学園短期大学の安房田司郎教授，高知植田病院の山本早智先生に深く感謝いたします．さらに，本論文の執筆の際，著者の研究室の院生，学部生にもお世話になりました．特に，院生の内田久也氏には多くのご協力をしていただきました．そして本論文の執筆の機会を与えてくださった高知工科大学に心からお礼申し上げます．

第 10 章　参考文献

(1) 内田久也・竹田史章：「ニューラルネットワークを用いた食事摂取量計測システムの構築」，システム制御情報学会講演論文集，pp.107~108 (2001)

(2) 竹田史章・内田久也・服部真一郎：「食事摂取量計測システムの構築」，高速信号処理応用技術学会 2001 年秋季研究会講演論文集，pp.75~80 (2001)

(3) H. Uchida and F. Takeda: “Development of a Measurement System for Food Intake using Neural Network”, The International Conference on Mechatronics and Information Technology ’01, pp.19~24 (2001)

(4) K. Fujiwara, F. Takeda, H. Uchida and Sakoobunthu Lalita: ”Dishes Extraction with Neural Network for Food Intake Measuring System”, SICE2002. (2002)

(5) 内田久也・竹田史章：「院内食事摂取量計測システムの開発」，Fuzzy, Artificial Intelligence, Neural Networks and Computational Intelligence Symposium ’02 in Saga，pp251-256(2002)

(6) 竹田史章・内田久也・安房田司郎・山本早智・長田司郎・小林茂樹・尾江俊昭：「院内用食事摂取量計測システムの開発」， 電気学会投稿中(2002)

(7) Takeda, F. and Omatu, S.: “Bank note recognition system using neural network with random masks”, Proceeding of the World Congress on Neural Networks, Portland, USA, Vol.1,pp.241~ 244 (1993)

(8) F. Takeda and S. Omatu: "Recognition System of US Dollars Using a Neural Network with Random Masks”,Proceedings of the International Joint Conference on Neural Networks, Nagoya, Japan, Vol.2,pp.2033- 2036 (1993)

(9) F. Takeda, S. Omatu, S. Onami, T. Kadono, and K. Terada:"A Paper Currency Recognition Method by a Small Size Neural Network with Optimized Masks by GA",

Proceedings of IEEE World Congress on Computational Intelligence, Orlando, USA, Vol.7,pp.4243- 4246 (1994)

(10) F. Takeda and S. Omatu: ”A Neuro-Money Recognition Using Optimized Masks by GA”, Advances in Fuzzy Logic, Neural Networks and Genetic Algorithms LNAI 1011 Springer pp.190-201 (1995)

(11) Takeda, F. and Omatu, S.: “high speed paper currency recognition by neural networks”, IEEE Transaction on Neural Networks, Vol.6,No.1, pp.73~77 (1995)

(12) F. Takeda and S. Omatu: ”A Neuro-System Technology for Bank Note Recognition”, Proceedings of Japan/USA symposium on Flexible Automation, Boston, USA,Vol.2,p.1511-1516 (1996)

(13) F. Takeda and S. Omatu: ” A Neruo-Recognition Technology for paper Currency Using Optimized Masks by GA and Its Hardware”, Proceedings of Inter- national Conference on Information Systems Analysis and Synthesis, Orlando, USA, pp.147-152 (1996)

(14) F. Takeda, S. Omatu and T. Nishikage: ” Neural Network Recognition System Tuned by GA and Design of Its Hardware by DSP”, Proceedings of Inter- national Symposium on Artificial Intelligence in Real-Time Control, Malaysia, pp.356-361 (1997)

(15) F. Takeda and S. Omatu: ”Neural Network Systems Technique and Applications In paper Currency Recognition”, Neural Network Systems, Techniques and Applications ACADEMIC Press Vol.5, Chapter 4, pp. 133-160 (1998)

(16) F. Takeda, S. Omatu and Y. Matsumoto:” Development of High Speed Neuro-Recognition Board and Application for Paper Currency”, The international Workshop on Signal Processing Application and Technology, pp.49-56 (1998)

(17) F. Takeda, T. Nishikage and Y. Matsumoto: ”Characteristic Extraction of Paper Currency

using Symmetrical Masks Optimized by GA and Neuro-Recognition of Multi-National Paper Currency", Proceedings of IEEE World Congress on Computational Intelligence, Alaska,USA,Vol.1,pp.634-639 (1998)

(18) T. Nishikage and F. Takeda: "Axis-Symmetrical Masks Optimized by GA for Neuro-Currency Recognition and their Statistical Analysis", Proceedings of WorldMulti-Conference on Systemics, Cybernetics, and Informatics, Orlando,USA,Vol.2,pp.308-314 (1998)

(19) 竹田史章, 西蔭紀洋, 内田久也, 中原昌樹, "紙幣用ニューロテンプレートマッチング識別手法の開発", 第 44 回システム制御情報学会論文誌, Vol.13, No.4, pp415-416, 2000

(20) Takeda, F., Nakahara, M., Ichiryu, Y., Uchida, H., "Autonomic Neuro- Recognition Board for Paper Currency", Proceeding of SPAT2000, pp85-90, 2000

(21) Yoshiko Ichiryu, Fumiaki Takeda, Hisaya Uchida and Masaki Nakahara : "Development of Neuro-Templates Matching Recognition Method of Banknotes", SCORED, pp60-61, 2001

(22) 内田久也, 竹田史章:「ニューラルネットワークを用いた米用色彩選別機開発」高速信号処理応用技術学会 2001 春季研究会, pp52－57, 2001

(23) 竹田史章, 一柳賀子, 内田久也, 中原昌樹, "米用色彩選別機へのニューロテンプレートマッチング識別手法の適用", 第 45 回システム制御情報学会研究発表講演会論文集, pp105-pp106, 2001

(24) Ichiryu, Y., Takeda, F., Uchida, H., Nakahara, M., et.al, "Proposal of a Rice Grading System Using Neuro-Template Matching Method", Proceeding of the Knowledge-Based Intelligent Information Engineering System & Allied Technologies, Vol2, pp.818-823, 2001

(25) Takeda, F., Uchida, H., et.al.: "A Proposal of Grading System for Fallen Rice Using

Neural Network”, WCCI2002, Honolulu, USA

(26)長友克介，竹田史章，中屋任喜，内田久也：「ニューラルネットワークを用いた落下米監査システムの開発」，電気学会産業計測制御研究会，pp53－58，2001

(27)末松良一，山田宏尚，「画像処理工学」，コロナ社，pp120-140，2000

(28)舟久保登，「パターン認識」，共立出版，pp154-157，1993

(29)安居院猛，長尾智晴，「C 言語による画像処理入門」，昭晃堂，pp47-74，2000

付録

ニューラルネットワーク

ニューラルネットワークとは，人間の脳の情報処理方式を計算機で実現しようというものである．したがって，人間の脳の神経細胞（ニューロン）をモデル化したものを多数つなげてネットワークを構成し，新しい情報処理原理を追求しようというものである．図１はニューロンをモデル化した様子である．

丸印で表されたのがニューロンで，各ニューロンは，入力層，中間層，出力層のそれぞれ独立した層に配置される．各層のニューロンはそれより１つ前の層のニューロンから入力を受ける．同じ層のニューロン間には結合がない．しかしながら，各層間ではすべてのニューロンが結びついているのが特徴である．

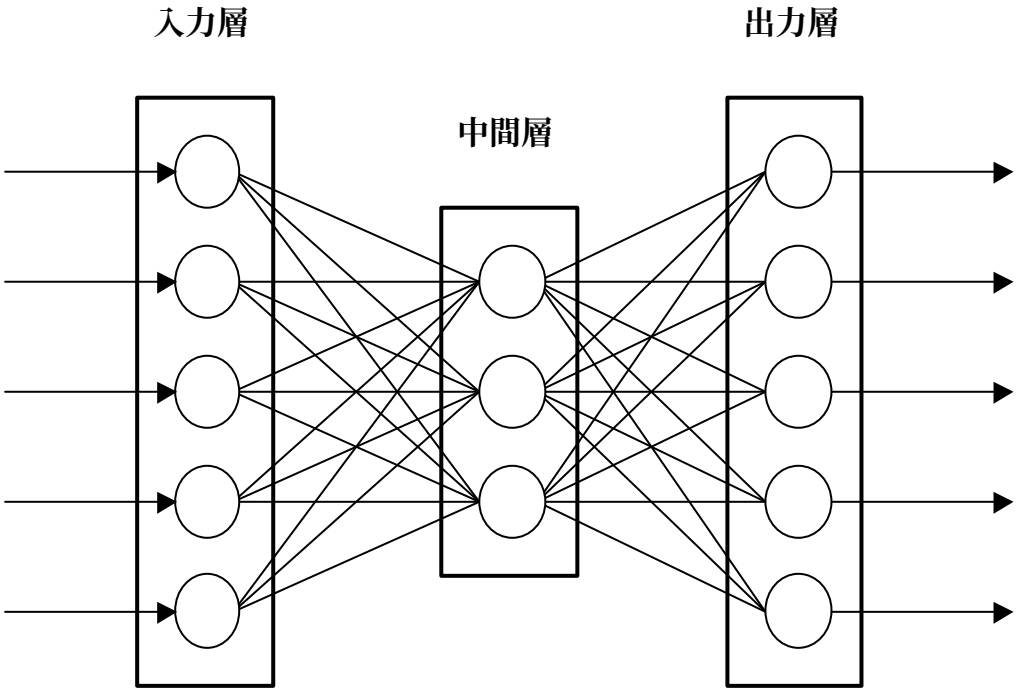


図１　ニューロンモデル

ニューラルネットワークの構成にはパターン認識に適した階層型ニューラルネッ

トワークと，連想記憶や組み合わせ最適問題に適した相互結合型ニューラルネットワークに分類できる．本研究では，食器パターン，食材残量パターンの識別を目的とするため，階層型ニューラルネットワークを使用する．そこで本節では，階層型ニューラルネットワークの情報伝達の方法と学習方法について説明する．図 2 はニューラルネットワークの情報伝達を示している．

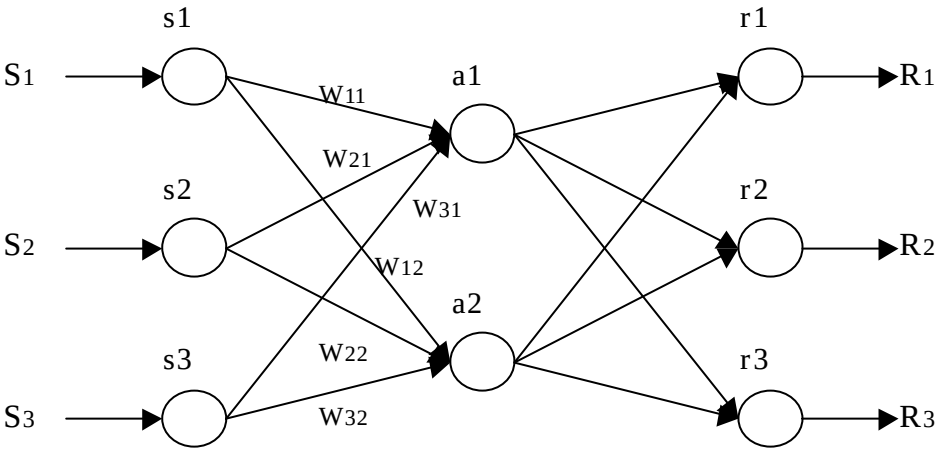


図２　ニューラルネットワークの情報伝播

情報は，入力パターンが入力層の各ニューロンに与えられ，中間層で変換されて，出力層の各ニューロンから出力パターンが得られる．図 2 で，s1，s2，s3 はそれぞれ入力層の第 1，2，3 ユニット，a1，a2 はそれぞれ中間層の第 1，2 ユニット，r1，r2，r3 はそれぞれ出力層の第 1，2，3 ユニットである．また，S_1，S_2，S_3 は入力層への入力値（スラブ値），R_1，R_2，R_3 は出力層の出力値を表す．また，W_{ij} は入力層の第 i ユニットから中間層の第 j ユニットへの重み（結合の強さ）である．a1 への入力の総和を I_1 とすると

$$I_1=W_{11}S_1+W_{21}S_2+W_{31}S_3 \qquad (1)$$

となり，a2 への入力の総和を I_2 とすると

$$I2=W12S1+W22S2+W32S3 \quad (2)$$

となる．つまり，一般的に中間層 j への入力の総和は

$$Ij=\sum_{i} WijSi \quad (3)$$

となる．また，中間層第 j ユニットの出力“ Oj ”はシグモイド関数を使用して式(4)で表される．式(4)において，θはユニット単位のしきい値である．T はネットワークの温度と呼ばれる正の数で，T が大きくなるほど図 3 に示すシグモイド関数のグラフはなだらかになる．

$$Oj=\frac{1}{1+\exp\left(\frac{-Ij+\theta}{1}\right)} \quad (4)$$

シグモイド関数は，出力がなめらかに変化しているため，微分しても連続な関数となり数学的に扱いやすいという大きな特徴がある．

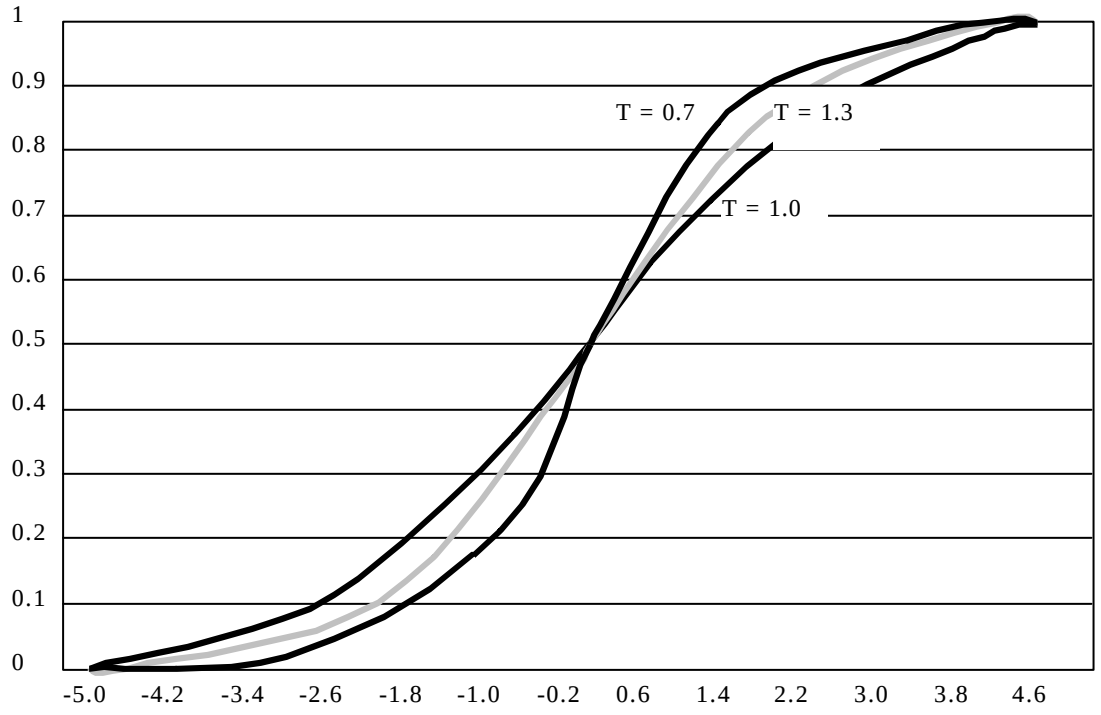


図 3　シグモイド関数

次に，学習方法について述べる．学習は，入力パターンに対して期待する出力パターンを得るために，各ユニット間の結合の重み（ウェイト）を適切な値に設定するための作業である．学習アルゴリズムには誤差逆伝播（Error Back Propagation）アルゴリズムを採用する．これは，いくつかの入力パターンの例（学習用データ）を与え，そのときの出力パターンと期待する出力パターン（教師値）との誤差が減少するようにウェイトを修正するアルゴリズムである．図 4 は誤差逆伝播アルゴリズムでの学習を表す．

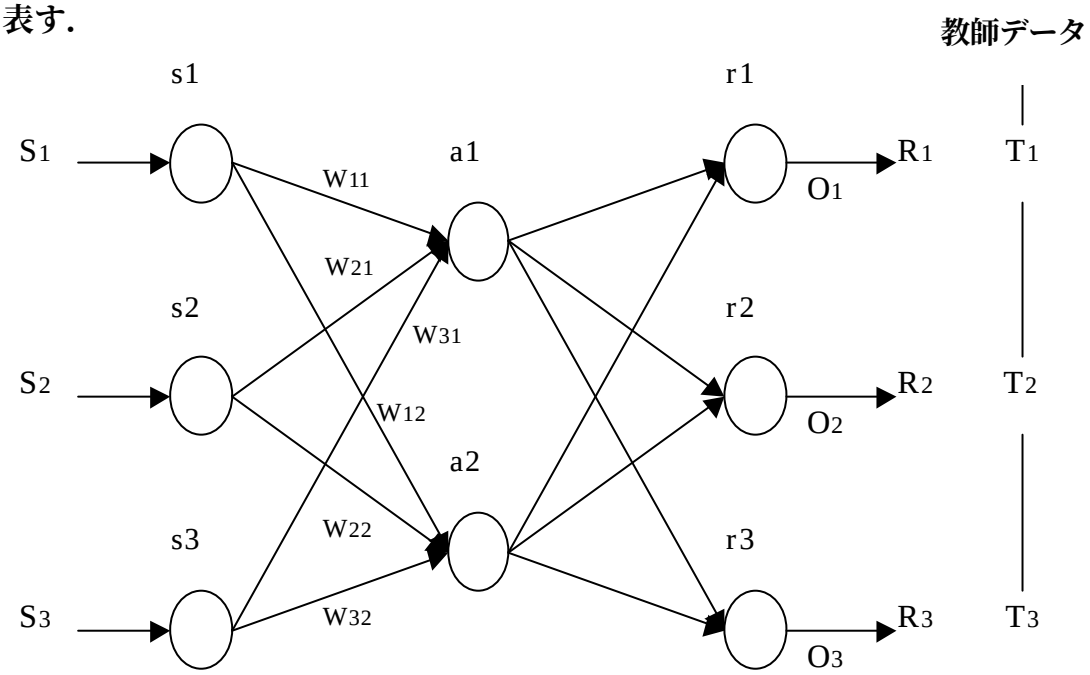


図 4　誤差逆伝播アルゴリズム

ある入力パターンを与えたときの出力層の第 j ユニットの出力値を Oj，このときの出力層の第 j ユニットの期待値（教師値）を Tj とすると，第 j ユニットの誤差 Ej は

$$Ej = \frac{1}{2}(Tj - Oj)^2 \qquad (5)$$

で定義される．また，1 つの学習パターン P における出力層の誤差を Ep とし，全学習パターンの誤差の総和を E とすると，Ep，E はそれぞれ式(2.6)，式(2.7)で表される．

E は総合誤差と呼ばれる．

$$Ep = \frac{1}{2}\sum_{j}(Ti - Oi)^2 \tag{6}$$

$$E = \sum Ep = \frac{1}{2}\sum_{p}\sum_{j}(Tj - Oj)^2 \tag{7}$$

誤差逆伝播アルゴリズムは，総合誤差 E が最小になるように各ユニット間のウェイトを修正する．修正の方法としては最急降下法を用い，各学習パターンごとに，各ユニットの誤差 Ej が最小になる方向へ微小な変更を加えていく．第 t 回目の学習における k－1 層の第 i ユニットから k 層の第 j ユニットへの重み Wij の修正量⊿Wij(k－1, k)(t)は式(8)で求まる．

$$\Delta W_{ij}^{k-1,k}(t) = -\varepsilon \delta_j^k O_i^{k-1} + \alpha \Delta W_{ij}^{k-1,k}(t-1) + \beta \Delta W_{ij}^{k-1,k}(t-2) \tag{8}$$

式(8)において，“ ε ”は学習定数，“ α ”は慣性定数，“ β ”は振動定数である．慣性定数とは，総合誤差の振動を減らし，学習の収束を加速させるための定数のことである．振動定数とは，総合誤差を上下に振動させて局所極小値（ローカルミニマム）から脱出させるための定数である．また学習定数とは，重みの修正量を調整するための定数である．“ δj(k) ”は k 層の第 j ユニットの一般化誤差で，k 層が中間層の場合と出力層の場合によって算出方法が異なる．まず，k 層が中間層の場合の一般化誤差を式(9)に示す．ただし，m は出力層のユニット番号を表す．

$$\delta_j^k = (\sum_m \Delta W_{jm}^{k,k+1} \delta_m^{k+1}) f'(I_j^k) \tag{9}$$

付録

食物除去アルゴリズムを反映させた食器抽出ソースコード

```
Option Explicit
Option Base 0

'------------------------------------------------------
Private Const PORT_NO As Integer = 20000
'------------------------------------------------------

Dim FileName1 As String
Dim FileName2 As String
Dim MenuF1Flag As Integer
Dim MenuF5Flag As Integer
Dim MenuF9Flag As Integer
Dim strArrivalData As String
Dim ImageFile As String
Dim Captured As Boolean


Private Sub CmdCapture_Click()
'*****
'*****   画像を 1 コマだけキャプチャする
'*****
    Dim X As Integer
    Dim Y As Integer
    Dim P_D As Long
    Dim P_Red As Integer
    Dim P_Green As Integer
    Dim P_Blue As Integer
    Dim P_Y As Integer

    Dim ret As Long
    Dim CapFileName As String

    If hWndC = 0 Then
        MsgBox "キャプチャ環境の初期化ができていません。" & vbCrLf & _
            "メニューのキャプチャ環境初期化を実施してください。", vbExclamation, "初期化未実施"
        Exit Sub
    End If
    ret = SendMessage(hWndC, WM_CAP_GRAB_FRAME, 0, 0&)
    ret = SendMessage(hWndC, WM_CAP_EDIT_COPY, 0, 0&)
    Picture2.Picture = Clipboard.GetData(vbCFPalette)
    Picture2.Picture = Clipboard.GetData(vbCFBitmap)
    '以下のコードを記述しないとキャプチャウィンドウ内の表示も停止する
'   ret = SendMessage(hWndC, WM_CAP_SET_OVERLAY, True, 0&)
    '保存ファイル名の自動生成
    CapFileName = MakeNewFileName(PlateDir, "Capture")
    ImageFile = CapFileName                    処理中のファイル名
    If CapFileName = "" Then
        MsgBox "キャプチャー画像を保管できませんでした。" & vbCrLf & _
            "キャプチャー画像を保管できるのは、最大 10000 枚です。" & _
            "既存のキャプチャー画像ファイルを削除してください。", vbExclamation, "画像保管エラー"
        Exit Sub
    End If
    '画像ファイルを保管
    SavePicture Picture2.Picture, PlateDir & CapFileName

    '濃淡画像に変換
    For X = 0 To 319
        For Y = 0 To 239
            PlateImage(X, Y) = Picture2.Point(X, Y)
        Next Y
    Next X

    For X = 0 To 319
        For Y = 0 To 239
            P_D = PlateImage(X, Y)
            P_Red = P_D Mod 256
            P_Green = Int(P_D / 256) Mod 256
            P_Blue = Int(P_D / 65536)
            P_Y = Int(CSng(P_Red) * 0.299 + CSng(P_Green) * 0.587 + CSng(P_Blue) * 0.114)
            DoEvents
            Picture2.PSet (X, Y), RGB(P_Y, P_Y, P_Y)
            PlateImage(X, Y) = P_Y
        Next Y
    Next X

    '画像ファイルリストボックスの更新
    Combo1.AddItem CapFileName
    Combo1.ListIndex = Combo1.NewIndex
    Captured = True
End Sub

Private Sub CmdEnd_Click()
    Dim Ans As Integer
    Ans = MsgBox('設定値を保存しますか？", vbYesNo)
    If Ans = 6 Then
        Call SaveSetFile
    End If
    Call SocketClose
    Unload Me
End Sub



Private Sub Form_Terminate()
    Call SocketClose
    Unload Me
End Sub

Private Sub FormTest_Click()
    frmPlateForm.Show vbModal
End Sub

Private Sub frmSet_Click()
    Settei.Show vbModal
```

```
    If  TestFlag = 1 Then
          Label13.Visible  =  True
           Label3(2).Caption = TestDate & " " &  TestTime
     Else
          Label13.Visible  =  False
           Label3(2).Caption = ""
    End  If
End Sub

Private Sub HELP_V_Click()
     MsgBox "Version 1.1.1"
End Sub

Private Sub DispNN_Click()
     NN_Conf.Show vbModal
End Sub

Private Sub Form_ Load()
     Dim  i As Integer
    Dim  j  As  Integer
     Dim  SaveFile As String
     Dim  strMessage As String

    If  App.PrevInstance Then
          strMessage = "このプログラムを複数起動することはできません。"
           MsgBox strMessage, vbExclamation
          Unload  Me
          Exit  Sub
     End  If

      Combo1.Clear
      List2.Clear
      Option1(0).Value = True
     Text5.Text  =  ""
      CmdRegist.Enabled = False

     For  i = 0 To 4
           Label7(i).Caption = ""
           Label8(i).Caption = ""
           Label9(i).Caption = ""
           Label10(i).Caption = ""
           Label12(i).Caption = ""
           Text1(i).Text  =  ""
           Text1(i).Visible = False
           Label4(i + 1).Caption = "-"
      Next i
      Label3(0).Caption = ""
      Label3(1).Caption = ""
      Label3(2).Caption = ""
      Label4(0).Caption = "-"
     Label14.Caption  =  ""
     Captured  =  False

      Call LoadSetFile

      '*****   01/09/04   *****
      'リストボックスの更新
      For  i = 0 To 9999
            SaveFile = "Capture" & Right("000" & i, 4) & ".bmp"
           If  Dir(SaveBmpDir & SaveFile) <> "" Then
                Combo1.AddItem  SaveFile
           End  If
      Next i

     If Combo1.ListCount > 0  Then
          Combo1.ListIndex  =  0
      End  If
      '**********************
      Call prm_read                          'NN 構造設定ファイル
      Call wload(WEIGHT_FILE)                'NN 重みファイル

      '*****   01/09/04   *****
      '2 値化画像ファイル保管ディレクトリの作成
     If  Dir(SaveBmpDir, vbDirectory) = "" Then
           MkDir (SaveBmpDir)
      End  If
      '**********************

      'キャプチャ環境の初期化
      Call InitCapSet

End Sub

Private Sub FileImageStart_Click()
     Dim  X  As Integer
     Dim  Y  As Integer

      ImageFile =  SaveBmpDir & Combo1.List(Combo1.ListIndex)
     Picture2.Picture  =  LoadPicture(ImageFile)
     Captured  =  False
    For  X  =  0  To  319
          For  Y  =  0  To  239
                DoEvents
                PlateImage(X, Y) = &HFF And Picture2.Point(X, Y)
           Next  Y
      Next  X
End Sub

'------------------------------------
Private Sub CmdStart_click()
     Dim  i As Integer
     Dim  PlateFile As String
     Dim  Thr As Integer
     Dim  Count1  As Integer
     Dim  Count2  As Integer
     Dim  Count3  As Integer
     Dim  Count4  As Integer
     Dim  ImageData As Long
     Dim  CenterX As Integer
     Dim  CenterY As Integer
    Dim  Distance  As Integer
     Dim  SaveFileName As String
```

```
    Dim LoadFileName As String

    Picture2.Cls
    Picture3.Cls
    Picture4.Cls
    Picture5.Cls
    Picture6.Cls
    Picture7.Cls
    For i = 0 To 4
        Label7(i).Caption = ""
        Label8(i).Caption = ""
        Label9(i).Caption = ""
        Label10(i).Caption = ""
        Label12(i).Caption = ""
        Text1(i).Text = ""
        Text1(i).Visible = False
        Label4(i + 1).Caption = "-"
    Next i
    Label3(0).Caption = ""
    Label3(1).Caption = ""
    Label3(2).Caption = ""
    Label4(0).Caption = "-"

    Captured = False

    SelfID = ""
    If Trim(Text5.Text) = "" Then
        MsgBox "個人 ID を入力してください。"
        Exit Sub
    End If

    CmdStart.Enabled = False

    SelfID = Trim(Text5.Text)              '個人 ID（MAX5 桁）
    List2.Clear
    List2.AddItem "食器問合せ・・"
    Call ReqPlateData
    If TestFlag = 1 Then
        Label3(2).Caption = MeasureDate & " " & Time()
    Else
        Label3(2).Caption = MeasureDate & " " & TestTime
    End If
    If PlateNo = 0 Then
        Exit Sub
    End If

    '代表食器画像の表示
    For i = 0 To PlateNo - 1
        Text1(i).Visible = True
        LoadFileName = StudyImageDir & "Plate" & Right("0" & Trim(Str(i + 1)), 2) & ".BMP"
        If Dir(LoadFileName) <> "" Then
            Picture1(i).Picture = LoadPicture(LoadFileName)
        End If
    Next i
'-------------------------------------------------
    '食物面積測定
    For i = 0 To PlateNo - 1
        Label4(0).Caption = (i + 1)
        List2.AddItem "食器 No." & Trim(Str(i + 1)) & " 食物面積演算開始"
        Call Recognize(i)
    Next i
    CmdRegist.Enabled = True
    CmdStart.Enabled = True
End Sub

Private Sub CmdRegist_Click()
    Dim Res As Integer
    Dim SaveFileName As String

    Res = MsgBox("測定結果を登録しますか？", vbYesNo)
    If Res <> 6 Then
        Exit Sub
    End If

    '測定結果を DB へ送信
    List2.AddItem "測定結果を DB へ送信・・"
    Call Response
'-------------------------------------------------
    If Captured = True Then
        List2.AddItem "測定画像ファイルの保管中・・"
        '*****  01/09/04  *****
        '2 値化画像ファイルの保管
        '保存ファイル名の自動生成
        SaveFileName = MakeNewFileName(SaveBmpDir, "CAPTURE")
        If SaveFileName = "" Then
            MsgBox "測定画像を保管できませんでした。" & vbCrLf & _
                   "測定画像を保管できるのは、最大 10000 枚です。" & _
                   "既存の画像ファイルを削除してください。", vbExclamation, "画像保管エラー"
            Exit Sub
        End If
        '画像ファイルを保管
        SavePicture Picture3.Image, SaveBmpDir & SaveFileName
        '画像ファイル名を表示
        List2.AddItem SaveFileName & " を保存"
    End If
    List2.AddItem "処理終了"
    Captured = False
End Sub

Private Sub Menu_InitCapture_Click()
    InitCapture.Show vbModal, Me
End Sub

Private Sub MenuSet_Click()
    Call SaveSetFile
End Sub

Private Sub ReqPlateData()
    Dim i As Integer
    Dim ReqCmdStr As String
    Dim SerialMinute As Integer
    Dim Item(100) As String
```

```
    Dim strWork As String
    Dim Posi As Integer
    Dim Posi1 As Integer
    Dim ItemCount As Integer

    If TestFlag = 1 Then
        SerialMinute = Hour(TestTime) * 24 + Minute(TestTime)
    Else
        SerialMinute = Hour(Time) * 24 + Minute(Time)
    End If
    If SerialMinute < 10 * 24 Then
        DayCode = "0"
        Label3(1).Caption = "朝食"
    ElseIf SerialMinute < 15 * 24 Then
        DayCode = "1"
        Label3(1).Caption = "昼食"
    Else
        DayCode = "2"
        Label3(1).Caption = "夕食"
    End If
    If TestFlag = 1 Then
        MeasureDate = Year(TestDate) & "/" & Right("0" & Month(TestDate), 2) & "/" & Right("0" & Day(TestDate), 2)
    Else
        MeasureDate = Year(Date) & "/" & Right("0" & Month(Date), 2) & "/" & Right("0" & Day(Date), 2)
    End If
    ReqCmdStr = "Com1," & SelfID & "," & MeasureDate & "," & DayCode

    Connect = 0
    RcvFlag = 0
    Call SocketConnect

    If Connect = 0 Then
        List2.AddItem "UDP ソケットエラー"
        Exit Sub
    End If

    Call SendUDP(ReqCmdStr)

    RcvFlag = 0
    Wait = 10
    TimeUp = 1

    While (TimeUp = 1 And RcvFlag = 0)
        DoEvents
    Wend
    If RcvFlag = 0 Then
        MsgBox "DBからのデータが受信できません"
        Exit Sub
    End If

    List2.AddItem "受信データ"
    List2.AddItem strArrivalData
    Posi = 1
    strWork = strArrivalData
    ItemCount = 0

    While (InStr(Posi, strWork, ",") > 0)
        Posi1 = InStr(Posi, strWork, ",")
        Item(ItemCount) = Mid(strWork, Posi, Posi1 - Posi)
        ItemCount = ItemCount + 1
        Posi = Posi1 + 1
    Wend
    Item(ItemCount) = Mid(strWork, Posi)

    PlateNo = Val(Item(0))
    Label3(0).Caption = PlateNo                    '食器の個数

    '-----------------------------------------
    '-- 食器の数を最大5に制限
    If PlateNo > 5 Then PlateNo = 5
    '-----------------------------------------
    For i = 0 To PlateNo - 1
        If Val(Item(i * 5 + 1)) = 0 Then
            Label7(i).Caption = "円形"
        Else
            Label7(i).Caption = "方形"
        End If
        PlateForm(i) = Val(Item(i * 5 + 1))      '食器の形状
        PlateLH(i) = Val(Item(i * 5 + 2))        '食器縦方向ピクセル数
        PlateLW(i) = Val(Item(i * 5 + 3))        '食器横方向ピクセル数
        STH(i) = Item(i * 5 + 4)                   '食器外縁検出スレシホールド値
        FTH(i) = Item(i * 5 + 5)                   '食器中食物面積測定検出スレシホールド値
    Next i

End Sub

Private Sub Response()
    Dim i As Integer
    Dim ResStr As String

    ResStr = "Com2," & SelfID & "," & MeasureDate & "," & DayCode & "," & Str(PlateNo)
    For i = 0 To PlateNo - 1
        ResStr = ResStr & "," & Text1(i).Text
    Next i

    Call SocketConnect

    If Connect = 0 Then
        List2.AddItem "UDP ソケットエラー"
        Exit Sub
    End If
    List2.AddItem "登録データ"
    List2.AddItem ResStr
    Call SendUDP(ResStr)

End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim i As Integer
    Dim RcvD() As Byte
    Dim DataLen As Integer
```

```
    Winsock1.GetData RcvD(), bytesTotal

    strArrivalData = ""
    For i = 0 To bytesTotal - 1
        strArrivalData = strArrivalData & Chr(RcvD(i))
    Next i
    RcvFlag = 1

End Sub


Private Sub SocketStatus()
    Select Case Winsock1.State
        Case sckClosed: Label14.Caption = "UDP Socket Closed": Connect = 0
        Case sckOpen: Label14.Caption = "UDP Socket Open": Connect = 1
        Case sckListening: Label14.Caption = "UDP Socket Listening"
        Case sckConnectionPending: Label14.Caption = "UDP Socket Connection Pending"
        Case sckResolvingHost: Label14.Caption = "UDP Socket Resolving Host"
        Case sckHostResolved: Label14.Caption = "UDP Socket Host Resolved"
        Case sckConnecting: Label14.Caption = "UDP Socket Connecting"
        Case sckConnected: Label14.Caption = "UDP Socket Connected"
        Case sckClosing: Label14.Caption = "UDP Socket Closing"
        Case sckError: Label14.Caption = "UDP Socket Error"
       Case Default: Label14.Caption = "Undefine State Code"
    End Select
End Sub

Private Sub SendUDP(DataStr As String)
    Dim SendB() As Byte
    Dim i As Integer

    ReDim SendB(Len(DataStr) - 1)

    For i = 0 To Len(DataStr) - 1
        SendB(i) = Asc(Mid(DataStr, i + 1, 1))
    Next i
    On Error Resume Next
    Winsock1.SendData SendB()

End Sub

Private Sub Winsock1_Error(ByVal Number As Integer, Description As String, ByVal Scode As Long, ByVal Source
As String, ByVal HelpFile As String, ByVal HelpContext As Long, CancelDisplay As Boolean)
    CancelDisplay = True
    MsgBox "WinSock Error:" & Description & "  Source:" & Source
End Sub

Private Sub Timer1_Timer()
    If Wait = 0 Then
        Exit Sub
    End If
    Wait = Wait - 1
    If Wait = 0 Then
        TimeUp = 0
    End If
End Sub
```

```
Private Sub SocketConnect()
    If Connect = 1 Then Exit Sub
    With Winsock1
        If .State = sckClosed Then
            .Protocol = sckUDPProtocol
            .RemoteHost = IP_ADR
            .LocalPort = PORT_NO + 1
            .RemotePort = PORT_NO
            .Bind
        End If
    End With
    Wait = 20
    TimeUp = 1
    While (TimeUp = 1)
        DoEvents
        If Winsock1.State = sckOpen Then
            TimeUp = 0
        End If
    Wend

    Call SocketStatus

End Sub


Private Sub SocketClose()
    Winsock1.Close
End Sub

'---------------------------------------------
Private Sub Recognize(PlateNo As Integer)
    Dim i As Integer
    Dim X As Integer
    Dim Y As Integer
    Dim U As Integer
    Dim V As Integer
    Dim X1 As Integer
    Dim Y1 As Integer
    Dim S As Integer
    Dim LH As Integer
    Dim LW As Integer
    Dim S_TH As Integer
    Dim F_TH As Integer
    Dim S_Image(160, 120) As Long   '食器切り出し用2値化画像データ
    Dim MonoD As Long
    Dim PlateX As Integer           '識別した食器のX中心座標（160*120 画像内）
    Dim PlateY As Integer           '識別した食器のY中心座標（160*120 画像内）
    Dim MAX_XPosi As Integer
    Dim MAX_YPosi As Integer
    Dim MAX_Value As Single
    Dim FoodCount As Integer
    Dim MAX As Single
    Dim outFoodFlg As Boolean '食材からでた
    Dim StatFlg As Boolean '食器位置にはいったかどうか
    Dim inFoodFlg As Boolean '食材に入った
```

付録

```
    Dim F_Ypar As Integer '食材Y座標の一番上
    Dim Y2 As Integer

    '----  食器の切り出しパラメータをセット
    S = PlateForm(PlateNo)              '食器形状（0:円形,1:方形）
    LH = PlateLH(PlateNo)               '食器縦ピクセル数
    LW = PlateLW(PlateNo)               '食器横ピクセル数
    S_TH = STH(PlateNo)                 '食器識別用スレシホールド値
    F_TH = FTH(PlateNo)                 '食物識別用スレシホールド値

    '---------------
    ' テスト用　パラメータ
'   S = 0
'   LH = 32
'   LW = 50
    '----------------------
    If S = 0 Then
        Label7(PlateNo).Caption = "円形"
    Else
        Label7(PlateNo).Caption = "方形"
    End If
    Label8(PlateNo).Caption = LH
    Label9(PlateNo).Caption = LW
    Label10(PlateNo).Caption = S_TH
    Label12(PlateNo).Caption = F_TH

    LH = LH * 2
    LW = LW * 2

    '----------------------

    '食器識別用2値化画像に変換し、Picture4、S_image(160,120)に画像を表示、格納
    Picture3.PaintPicture Picture2.Picture, 0, 0, 160, 120, 0, 0
    For X = 0 To 159
        For Y = 0 To 119
            If (&HFF And Picture3.Point(X, Y)) > S_TH Then
                MonoD = &HFFFFFF
            Else
                MonoD = 0
            End If
            S_Image(X, Y) = MonoD
            Picture4.PSet (X, Y), MonoD
        Next Y
    Next X

    MAX_Value = -1


    '-------------------------------------------------------------------
    If S = 0 Then           '円形
        For X = LW / 2 To 159 - (LW / 2) Step StepC
            For Y = LH / 2 To 119 - (LH / 2) Step StepC
                Picture4.PSet (X, Y), vbRed
                Picture5.Cls
                For X1 = 0 To 63
                    For Y1 = 0 To 63
                        MImage(X1, Y1) = 0
                    Next Y1
                Next X1
                For X1 = -(LW / 2) To LW / 2
                    StatFlg = False '食器の上のふちかどうか
                    outFoodFlg = False '
                    inFoodFlg = False '食器から食材に入った
                    F_Ypar = 0
                    DoEvents
                    For Y1 = -(LH / 2) To LH / 2
                        If (CSng(X1 * X1) / CSng(LW * LW / 4) + CSng(Y1 * Y1) / CSng(LH * LH / 4)) <= 1
Then
                            If (CSng(X1 * X1) / CSng(LW * LW / 4) + CSng(Y1 * Y1) / CSng(LH * LH / 4))
>= CSng(Inner / 100) Then

                                '藤原追加
                                If Picture4.Point(X + X1, Y + Y1 - 1) = 0 Then '一つ前が黒ならば
                                    If Picture4.Point(X + X1, Y + Y1) = &HFFFFFF Then '現在の位置が
白ならば
                                        If StatFlg = True Then '食器の上のふちでないならば
                                            If outFoodFlg = False Then '食材の外ないならば
                                                F_Ypar = Y1
                                                inFoodFlg = True
                                            End If
                                        End If
                                    End If
                                End If
                                If inFoodFlg = True Then
                                    If Picture4.Point(X + X1, Y + Y1 + 1) = 0 Then
                                        If (CSng(X1 * X1) / CSng(LW * LW / 4) + CSng((Y1 + 1) * (Y1 +
1)) / CSng(LH * LH / 4)) < 1 Then
                                            For Y2 = F_Ypar To Y1
                                                MImage(32 + X1 / 2, 32 + Y2 / 2) = 0
                                                Picture5.PSet (32 + X1 / 2, 32 + Y2 / 2), MImage(32 +
X1 / 2, 32 + Y2 / 2)
                                                outFoodFlg = True
                                            Next Y2
                                        End If
                                    End If
                                End If
                                MImage(32 + X1 / 2, 32 + Y1 / 2) = Picture4.Point(X + X1, Y + Y1)

                                If Option1(0).Value = True Then
                                Picture5.PSet (32 + X1 / 2, 32 + Y1 / 2), MImage(32 + X1 / 2, 32 + Y1 /
2)
                                End If
                            End If
                            StatFlg = True
                        End If
                    Next Y1
                Next X1
                If MAX_Value < PlateRecognize(PlateNo) Then
                    MAX_Value = PlateRecognize(PlateNo)
                    MAX_XPosi = X
                    MAX_YPosi = Y
                End If
```

```
                Label4(PlateNo + 1).Caption = Format(MAX_Value, "0.00")
            Next Y
        Next X
    Else                    '方形
      For X = LH / 2 To 159 - (LH / 2) Step StepC
        For Y = LW / 2 To 119 - (LW / 2) Step StepC
            Picture4.PSet (X, Y), vbRed
            Picture5.Cls
          For X1 = 0 To 63
              For Y1 = 0 To 63
                  MImage(X1, Y1) = 0
              Next Y1
            Next X1
        For X1 = -(LW / 2) To LW / 2
            DoEvents
          For Y1 = -(LH / 2) To LH / 2
              If (Abs(X1) >= ((Inner * LW / 2) / 100)) Or (Abs(Y1) >= ((Inner * LH / 2) / 100)) Then
                  MImage(32 + X1 / 2, 32 + Y1 / 2) = Picture4.Point(X + X1, Y + Y1)
                If Option1(0).Value = True Then
                Picture5.PSet (32 + X1 / 2, 32 + Y1 / 2), MImage(32 + X1 / 2, 32 + Y1 / 2)
                End If
              Else
                LH = LH
              End If
            Next Y1
          Next X1
          If MAX_Value < PlateRecognize(PlateNo) Then
            MAX_Value = PlateRecognize(PlateNo)
            MAX_XPosi = X
            MAX_YPosi = Y
          End If
          Label4(PlateNo + 1).Caption = Format(MAX_Value, "0.00")
        Next Y
      Next X
    End If

    Picture5.Cls
    Picture6.Cls
  For X1 = -(LW / 2) To LW / 2
      StatFlg = False '食器の上のふちかどうか
      outFoodFlg = False '
      inFoodFlg = False '食器から食材に入った
      F_Ypar = 0
      DoEvents
    For Y1 = -(LH / 2) To LH / 2
        If S = 0 Then
            If (CSng(X1 * X1) / CSng(LW * LW / 4) + CSng(Y1 * Y1) / CSng(LH * LH / 4)) <= 1 Then
                If (CSng(X1 * X1) / CSng(LW * LW / 4) + CSng(Y1 * Y1) / CSng(LH * LH / 4)) >=
CSng(Inner / 100) Then

                    '藤原追加
                    If Picture4.Point(MAX_XPosi + X1, MAX_YPosi + Y1 - 1) = 0 Then '一つ前が黒な
らば

                        If Picture4.Point(MAX_XPosi + X1, MAX_YPosi + Y1) = &HFFFFFF Then '現
在の位置が白ならば

                            If StatFlg = True Then '食器の上のふちでないならば

                                If outFoodFlg = False Then '食器の上のふちでないならば
                                    F_Ypar = Y1
                                    inFoodFlg = True
                                End If
                            End If
                        End If
                    End If
                    If inFoodFlg = True Then
                        If Picture4.Point(MAX_XPosi + X1, MAX_YPosi + Y1 + 1) = 0 Then
                            If (CSng(X1 * X1) / CSng(LW * LW / 4) + CSng((Y1 + 1) * (Y1 + 1)) /
CSng(LH * LH / 4)) < 1 Then
                                For Y2 = F_Ypar To Y1
                                    'MImage(32 + X1 / 2, 32 + Y2 / 2) = 0
                                Picture5.PSet (32 + X1 / 2, 32 + Y2 / 2), 0 'MImage(32 + X1 / 2, 32
+ Y2 / 2)
                                    outFoodFlg = True
                                Next Y2
                            End If
                        End If
                    End If

                      Picture5.PSet (32 + X1 / 2, 32 + Y1 / 2), Picture4.Point(MAX_XPosi + X1,
MAX_YPosi + Y1)
                    StatFlg = True
                End If
            End If
        Else
            If (Abs(X1) >= (Inner * LW / 2 / 100)) Or (Abs(Y1) >= (Inner * LH / 2 / 100)) Then
             Picture5.PSet (32 + X1 / 2, 32 + Y1 / 2), Picture4.Point(MAX_XPosi + X1, MAX_YPosi +
Y1)
            End If
        End If
    Next Y1
  Next X1


    '食物面積を測定
  For X = 0 To 159
      For Y = 0 To 119
          If (&HFF And Picture3.Point(X, Y)) > F_TH Then        '食物スレシホールド値で２値化
              MonoD = &HFFFFFF
          Else
              MonoD = 0
          End If
          S_Image(X, Y) = MonoD
          Picture7.PSet (X, Y), MonoD
      Next Y
  Next X


    '識別食器範囲を表示
  If S = 0 Then    円
      MAX = LW
    If MAX < LH Then MAX = LH
      Picture7.Circle (MAX_XPosi, MAX_YPosi), MAX / 2, vbBlue, , , LH / LW
      Picture3.Circle (MAX_XPosi, MAX_YPosi), MAX / 2, vbBlue, , , LH / LW
```

```
        Picture4.Circle (MAX_XPosi, MAX_YPosi), MAX / 2, vbBlue, , , LH / LW
      If Inner <> 0 Then
          Picture7.Circle (MAX_XPosi, MAX_YPosi), ((MAX * Inner) / 2) / 100, &HFFC0C0, , , LH / LW
          Picture3.Circle (MAX_XPosi, MAX_YPosi), ((MAX * Inner) / 2) / 100, &HFFC0C0, , , LH / LW
          Picture4.Circle (MAX_XPosi, MAX_YPosi), ((MAX * Inner) / 2) / 100, &HFFC0C0, , , LH / LW
      End If
   Else                方形
      Picture7.Line (MAX_XPosi - LW / 2, MAX_YPosi - LH / 2)-(MAX_XPosi + LW / 2, MAX_YPosi + LH /
2), vbBlue, B
      Picture3.Line (MAX_XPosi - LW / 2, MAX_YPosi - LH / 2)-(MAX_XPosi + LW / 2, MAX_YPosi + LH /
2), vbBlue, B
      Picture4.Line (MAX_XPosi - LW / 2, MAX_YPosi - LH / 2)-(MAX_XPosi + LW / 2, MAX_YPosi + LH /
2), vbBlue, B
      If Inner <> 0 Then
          Picture7.Line (MAX_XPosi - (LW * Inner / 2) / 100, MAX_YPosi - (LH * Inner / 2) / 100)-
(MAX_XPosi + (LW * Inner / 2) / 100, MAX_YPosi + (LH * Inner / 2) / 100), &HFFC0C0, B
          Picture3.Line (MAX_XPosi - (LW * Inner / 2) / 100, MAX_YPosi - (LH * Inner / 2) / 100)-
(MAX_XPosi + (LW * Inner / 2) / 100, MAX_YPosi + (LH * Inner / 2) / 100), &HFFC0C0, B
          Picture4.Line (MAX_XPosi - (LW * Inner / 2) / 100, MAX_YPosi - (LH * Inner / 2) / 100)-
(MAX_XPosi + (LW * Inner / 2) / 100, MAX_YPosi + (LH * Inner / 2) / 100), &HFFC0C0, B
      End If
   End If

   FoodCount = 0
  If S = 0 Then    円
     For X = MAX_XPosi - LW / 2 To MAX_XPosi + LW / 2
        For Y = MAX_YPosi - LH / 2 To MAX_YPosi + LH / 2
           If (CSng(X - MAX_XPosi) * CSng(X - MAX_XPosi) / CSng(LW * LW / 4) + CSng(Y -
MAX_YPosi) * CSng(Y - MAX_YPosi) / CSng(LH * LH / 4)) <= CSng(Inner / 100) Then
              If (&HFF And Picture7.Point(X, Y)) >= 255 Then
                 FoodCount = FoodCount + 1
              End If
           End If
        Next Y
     Next X
   Else                方形
     For X = MAX_XPosi - (Inner * LW / 2) / 100 To MAX_XPosi + (Inner * LW / 2) / 100
        For Y = MAX_YPosi - (Inner * LH / 2) / 100 To MAX_YPosi + (Inner * LH / 2) / 100
           If (&HFF And Picture7.Point(X, Y)) >= 255 Then
              FoodCount = FoodCount + 1
           End If
        Next Y
     Next X
  End If
   Text1(PlateNo).Text = FoodCount

End Sub
```

次に，k 層が出力層の場合の一般誤差化を式(10)に示す．Ij(k)は k 層の第 j ユニットの総和を表す．

$$\delta_j^k = (T_j - O_j^k) f'(I_j^k) \tag{10}$$

以上が誤差逆伝播法によるウェイト修正の概略である．

上記の構成で設計されるニューラルネットワークはコンピュータが不得意としていた非線形識別であるパターン認識や，学習，自己組織化などを得意とし，様々な分野でその技術が応用されている．