

令和5年度卒業論文

組み込み機器上で動作する深層学習を用いた
低消費電力顔認証システムの検討

宮崎大学情報システム工学科

森茂蒼士

指導教員 椋木雅之教授

概要

本論文では、ArcFace を用いた顔認証システムにおいて、深層学習のバックボーンを変えて、その時の消費電力を測定し比較実験をすることで、より効率的な顔認証システムを提案する。ArcFace とは顔認識技術の一種であり、同じく顔認証技術の一つである FaceNet が距離に着目しているのに対して、ArcFace は角度に着目している。本研究は顔認証システムを構築するにあたり、低コスト小型のシングルボードコンピュータである Raspberry Pi 及び専用のカメラモジュールを用いる。Raspberry Pi を用いた顔認証についての研究はこれまでも行われてきているが、精度や計算量に着目したものが多くシステム全体として消費電力の削減を目指したものはまだ少ない。

本研究では、ArcFace のバックボーンを ResNet18、ResNet50、MobileNet の 3 つに変えて、データセットに VGGFace2 を用いて学習を行う。出来たモデルを Raspberry Pi で用いて顔認証システムを構築し、精度、速度、消費電力の 3 つの観点から各パラメータを決めることにより効率的なシステムを提案する。

目次

1.	はじめに	1
2.	組み込み機器上で動作する顔認証システム	3
2.1	顔認証システムの必要性	3
2.2	深層学習を使った顔認証手法	3
2.3	効率的な顔認証システムの従来研究	5
2.4	Raspberry Pi	5
2.5	ArcFace	8
2.6	ResNet と MobileNet	9
3.	低消費電力顔認証システムの設計	10
3.1	消費電力を考慮した顔認証システム設計	10
3.2	設計方針	11
3.3	消費電力を考慮した顔認証システムにおけるパラメータの設定	12
4.	要素技術の評価	15
4.1	バックボーンを変えた ArcFace による学習結果	15
4.2	顔検出の精度評価と消費電力測定の結果	16
4.3	バックボーンを選定	18
4.4	顔認証の際の閾値の決定	20
5.	設計した顔認証システム	23
6.	結論	27
	謝辞	28
	参考文献	29

1. はじめに

現在、指紋や顔などの生体情報を用いた生体認証が様々な分野で広く用いられている[1]。モバイルデバイスでは顔画像を用いた、顔認証が一般的となっている。従来の顔認証システムは通常高い計算リソースを必要とし、それが消費電力の増加に繋がっていた。しかし、最近の深層学習の進展により、効率的で高性能なモデルが提案され、組み込み機器向けの適用が可能となった[2]。組み込み機器の普及と共に、低消費電力かつ高性能な顔認証システムの需要がますます増加している。

本研究では、深層学習技術を活用し、組み込み機器上で動作する低消費電力の顔認証システムに焦点を当てる。

顔認証に用いる深層学習として、本研究では ArcFace[3]を用いる。ArcFace において、学習時には、特徴抽出を行う深層ネットワークであるバックボーンの前には、顔識別を行うヘッドを加えたネットワーク構造を用いる。このネットワークのバックボーンで得た特徴量を長さ 1 のベクトルに正規化し、ヘッド部分で個人ごとに分類する。その正誤に基づき、ArcFace 独自の損失関数を最小化することで、バックボーンの重みを決定していく。顔認証時には、上記により学習済みのバックボーン部分のみを用いて、与えられた顔画像から特徴抽出を行い、得られた特徴ベクトル間の角度により類似度を計算して、認証を行う。

特徴量抽出を行うバックボーンには、既存の様々なネットワークを用いることができる。例えば、より精度を上げる目的で作られた ResNet50[4]やリソースが少ない環境を想定して作られた MobileNet[5]などが用いられる。Xu ら [6]

は顔認証技術の一つである FaceNet のバックボーンを MobileNet にして実装した研究を行っている。また、Dang ら[7]は、ArcFace のバックボーンを MobileNet にして実装し IoT へ応用することの利点を主張した。だが、これらは顔認証のための深層学習部分の効率化については評価しているが、組み込み機器上で動作させた際の電力消費量に基づき、システム全体が低消費電力となるように設計した研究事例は見当たらない。

本研究では、ArcFace のバックボーンに既存の様々な深層学習ネットワークを用いてそれらを比較実験し、消費電力を計測する。その結果に基づいて、バックボーンや認証における様々なパラメータを決定し、組み込み機器上で動作する低消費電力顔認証システムを設計する。具体的にはバックボーンとして高い識別性能を持つ ResNet50、ResNet の構造を簡略化して効率を高めた ResNet18、性能を保ちつつ効率に焦点を当てて簡略な構造を採用した MobileNet を比較する。また、システム設計のパラメータとして、認証回数や検出の間のスリープ時間を決定する。

以下、2 章では、顔認証システムの必要性と顔認証に関する従来研究を述べ、本研究で用いる要素技術や組み込み機器について説明をする。3 章では、低消費電力顔認証システムとして本研究における目標値とパラメータを設定し、そのための設計方針を決定する。4 章では、3 章の設計方針に基づいて本研究における顔認証システムで用いる各要素技術の評価を行う。5 章では、4 章の結果をもとに 3 章で設定した目標値を満たすように顔認証システムにおけるパラメータを設定する。最後に 6 章では、結論と今後の課題について述べる。

2. 組み込み機器上で動作する顔認証システム

2.1 顔認証システムの必要性

現在、指紋など個人に固有の生体情報を用いる生体認証が広く用いられている。顔画像を用いた顔認証もまた、多く用いられている生体認証のうちの一つである。顔認証の強みは、特別なセンサーや機器を必要とせず、カメラのみで認証が可能な点である。また、顔認証は物理的な接触が少なく、非接触での認証が可能のため、感染症対策としても期待されている。

顔認証に用いる深層学習は通常多くの計算リソースを必要とする。そのため、少ない計算リソースしか持たない組み込み機器で利用することは困難であった。しかし、最近の深層学習の進展により、効率的で高性能なモデルが提案され、計算リソースの少ない機器での利用が可能となった。それに伴い、低消費電力かつ高性能な顔認証システムの需要がますます高まっている。

2.2 深層学習を使った顔認証手法

顔認証を行う際には、深層学習を用いることが主流となっている。深層学習は、多層のニューラルネットワークを使用して複雑なパターンや特徴を学習することができ、その高い表現力と柔軟性から、顔認証において優れた性能をもつ。深層学習ではまず、与えられた学習用の画像に対して畳み込みを行う。畳み込みとは、画像の特徴を強調するためのフィルターをかけることであり、フィルターには重みが決められている。次に、畳み込みで得られた特徴量に非線形性を持たせるために、活性化関数を適用する。その後、プーリング層で計算コストを削減し

つつ、適切に画像の特徴の抽象化を行う。そして、出力層で畳み込みとプーリングを繰り返して得られた特徴量をもとに入力画像を既存のカテゴリに分類する。学習時には、損失関数を用いて、出力層で得た予測値と正解との誤差を求める。予測が正しければ、損失関数の値は小さくなる。損失関数の結果に応じてフィルターの重みを適切に変えていくことで損失関数の値を小さく、つまり予測値と正解との誤差を小さくする。深層学習において最終的な目的は、これらを繰り返すことで正確な予測を行うための最適なネットワークの重みを得ることにある。

本研究では、顔認証に ArcFace という深層学習を用いる。ArcFace では学習する際には、特徴抽出を行う深層ネットワークであるバックボーンの前に、顔識別を行うヘッドを加えたネットワーク構造をしている(図1)。特徴量抽出を行うバックボーンには、既存の様々なネットワークを用いることができる。

学習時



認証時



図1 バックボーンとヘッドの関係

2.3 効率的な顔認証システムの従来研究

効率的な顔認証システムの従来研究として、Xu ら[6]の研究では顔認証技術の一つである FaceNet のバックボーンとして MobileNet と ResNet を使用したものを比較実験している。この研究では結果的に MobileNet を用いたことで計算コストを ResNet 使用時の半分にしている。さらに、Xu らは独自の乗数パラメータを導入してネットワークを均一に薄くすることで、認証精度をほとんど落とさずに、計算コストを3分の1まで削減している。また、Dang ら[7]の研究では本研究と同様にバックボーンに MobileNet を用いた ArcFace を作成しているが、評価指標として精度に着目している。Fushuai Wang ら[8]の研究では、MobileNetV2 を改良し、顔認識と分類に ArcFace を使用し、本研究でも使用する Raspberry Pi 4B で顔分類プログラムをテストしている。この研究では、最終的にテスト用の画像で 94.4%、学習用の画像で 99.8%の精度を出したと結論づけられていた。

このように ArcFace を計算リソースの少ない組み込み機器上で動作させるためにバックボーンに MobileNet を用いて検証した先行研究はあるが、それらの多くは精度や計算コストを評価項目としたものである。本研究では、顔認証システムを組み込み機器上で動作させた際の電力消費量を測定することで、消費電力の少ない顔認証システムを設計する。

2.4 Raspberry Pi

本研究では、顔認証システムに用いる組み込み機器として Raspberry Pi[9](図

2)を用いる。Raspberry Pi とは ARM プロセッサを搭載したシングルボードコンピュータである。イギリスの Raspberry Pi 財団によって開発されている。MoPi という専用のキットを用いることで電池でも動かすこともでき、消費電力も少ない。また、専用のカメラモジュールや Bluetooth、Wi-Fi、HDMI ケーブルも接続可能であり、低コストでありながら高いパフォーマンスと柔軟性を備えている。

Raspberry Pi には、Raspberry Pi zero、Raspberry Pi Pico、Raspberry Pi Model A、Raspberry Pi Model B などいくつかのモデルがある。これらには、CPU 性能、価格、機能、大きさなどの違いがある。本研究ではその中でも、最も高性能なモデルである Raspberry Pi Model 4B を使用する。Raspberry Pi Model 4B の性能を表 1 に示す。

また本研究では、Raspberry Pi のカメラ入力には Raspberry Pi 専用のカメラモジュールを用いる。このカメラモジュールは、画素数が 800 万、静止画像解像度が 3280 x 2464 ピクセルであり、フレームレートは 1080p で 30fps となっている。

さらに、本研究では、電流計測に Raspberry Pi 専用の電流計測装置である RPZ-PowerMGR (図 3)を使用する。RPZ-PowerMGR ではターミナルでコマンドを入力することにより、指定した時間から最大 1 時間、毎秒の消費電流値を測定することができる。



図 2 Raspberry Pi Model 4B

(<https://www.iodata.jp/product/pc/raspberrypi/ud-rp4b/spec.htm>)

表 1 Raspberry Pi Model 4B の性能

	Raspberry Pi Model 4B
CPU	クアッドコア 1.5GHz
GPU	デュアルコア VideoCore IV 500MHz
メモリ	8GB
有線 LAN	○
無線 LAN	○
Bluetooth	Bluetooth 5.0 BLE
USB2.0	2 個
USB3.0	2 個
HDMI 出力	マイクロ HDMI 2 個(4K 対応)
推奨電源 USB	Type-C 5V3A
サイズ(mm)	85×56×17mm(幅×奥×高さ)
GPI コネクタ	40-ピン GPIO
OS	Linux 6.1.21-v8+

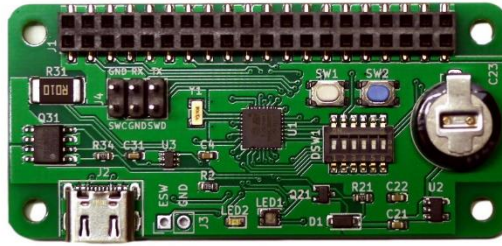


図 3 RPZ-PowerMGR

(<https://www.indoorcorgielec.com/2021/11/05/rpz-powermgr-rev2-release/>)

2.5 ArcFace

ArcFace[3]は顔認証技術のひとつである。ArcFace では式(1)に示す Additive Angular Margin Loss という損失関数が用いられている。

$$-\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s(\cos(\theta_{y_i+m}))}}{e^{s(\cos(\theta_{y_i+m}))} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}} \quad (1)$$

ここで、 y_i はサンプル*i*に対する真のクラスラベル、 θ_j は、サンプル*i*の特徴ベクトルと*j*番目のクラスの中心ベクトルとのなす角度、 N はミニバッチ内のサンプル数であり、 n は学習時のクラス数である。また s はスケールパラメータ、 m はマージンパラメータであり学習時に定めるハイパーパラメータである。この損失関数の目的は、特徴ベクトルのクラス内分散を小さくし、クラス間分散を大きくするような学習を行うことにある。つまり、同じクラスの特徴ベクトル同士の角度を小さくし、異なるクラスの特徴ベクトル同士の角度を大きくすることで、クラス間の識別性を高めることである。

顔認証時には、2つの顔画像 A,B から抽出した特徴ベクトル A、B に対して、

式(2)のコサイン類似度を求める。コサイン類似度では、2つのベクトルの類似性を表す値が-1から1の間で出力される。値が、-1に近いほど2つのベクトルの類似性が低く、1に近いほど類似していることを表す。コサイン類似度が、あらかじめ定められた閾値 δ 以上であれば、顔画像A、Bは同一人物、 δ 未満であれば別人物と判定する。

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} \quad (2)$$

2.6 ResNet と MobileNet

ResNet[4]や MobileNet[5]は ArcFace の深層学習においてバックボーンで用いられるネットワーク構造の代表的なモデルである。ResNet は精度を上げるために畳み込み層が深くなっている。ResNet50 は層が深いためより複雑な特徴やパターンを学習することが可能であるが、その分計算コストが高くなる。ResNet18 は ResNet50 に比べて層が浅い分精度は落ちるが、比較的軽量でありリアルタイム性が重視されている場面での使用に適している。MobileNet は、主にリソースに制約のあるデバイスやモバイルアプリケーションなどでの実行を目的として開発された。そのため畳み込み層における計算を効率的に行うことができ、高い性能を維持しつつも、モデルのサイズと計算量を最小限に抑えることができる。つまり、Raspberry Pi のようなリソースの限られた組み込み機器では MobileNet の方が適していると言える。本研究では、この違いがどの程度消費電力としてシステムに影響を与えるのかを測定し、システム全体の消費電力を抑えた最適な運用方法を検討する。

3 低消費電力顔認証システムの設計

3.1 消費電力を考慮したシステム設計

本研究では消費電力を考慮した顔認証システムを構築する。顔認証を行う状況と目標値を次のように設定する。

〈状況〉

プログラムを実行した状態で Raspberry Pi を研究室入口に設置して、入室の際に1時間あたり25人程度が利用する。認証できた場合は入室を許可とする。

〈目標値〉

消費電力	消費電力の平均値を Raspberry Pi 起動のみの 120%以下にする
平均認証待ち時間	認証にかかる時間を平均 6 秒以内とする
他人受入率	他人受入率を 0.1%未満とする

消費電力は低いほど良いが、入室者が近づいてきたことを検知する処理を常に行う必要があることを考慮して、起動時のみの 120%以下とした。平均認証待ち時間は、入室者が歩いてきてカメラに写ってから、カメラの前にくるまでを3秒程度として、その後カメラの前で3秒程度待機することを想定している。他人受入率は、誤って部外者が入室できてしまうことを防ぐために低めの値として 0.1%未満を目標値とする。

この目標値を満たすために、まずは各モデルの学習を行い性能を比較する。そして、Raspberry Pi 上で ResNet18, ResNet50, MobileNet をバックボーンとして学習した ArcFace を用いて顔認証を行う。また、認証中の電流値を Raspberry

Pi 専用の電流計測装置である RPZ-PowerMGR により計測する。本来、電力は電流と電圧の積により求めることができるが、Raspberry Pi の電圧は約 5V で一定であるため、本研究では電流値を指標にして、電流値を下げることで消費電力の削減を目指す。従来研究では顔認証システムにおいて、精度や計算コストに焦点を当てているものが多かったが、本研究ではそれぞれ同等の精度のモデルを比較実験して、顔検出に一定のスリープ時間を入れ、顔認証の回数を設定することで消費電力を抑えた最適な運用方法を検討する。

顔認証システムを構築するにあたり、カメラ内に顔が映った場合は、検出し切り出す必要がある。本研究では、カメラ入力に Raspberry Pi 専用のカメラモジュールを使用して、顔検出に OpenCV Haar-Cascade[10]を用いる。

3.2 設計方針

顔認証システムを構築するに伴い、システムに用いる各要素技術について次のように評価を行う。

1. バックボーンを変えた ArcFace による学習

ResNet18, ResNet50, MobileNet をバックボーンとして ArcFace による学習を行う。学習には VGGFace2[11]の顔画像のデータを用いる。VGGFace2 とは顔認識や顔検出のためのデータセットであり、9131 人の約 330 万枚の画像がある。学習後は各モデルの精度に関して比較を行う。

2. 顔検出の精度評価と消費電力測定

OpenCV Haar-Cascade を用いた顔検出の精度評価と、顔検出 1 回あたりの平均検出時間、消費電力を求める。同時に、Raspberry Pi 起動のみの消費電力の測定も行う。

3. バックボーンを選定

Raspberry Pi 上で ResNet18, ResNet50, MobileNet をバックボーンとして学習した ArcFace を用いてそれぞれ顔認証を行い、RPZ-PowerMGR により電流値を計測する。顔認証は同等の精度のモデルを用いて行い、登録している顔画像との比較を行う。また、モデルごとの認証一回当たりの消費電力を求める。加えて、認証一回にかかる平均認証時間、平均モデル処理速度を求める。これらすべてを踏まえて、顔認証システムに用いるモデルを決定する。

4. 顔認証の際の閾値の決定

上記で選定されたモデルに関して、認証の際に入出を許可する際の閾値を決定する。その為に、VGGFace2 の検証用の画像を用いて、本人同士と他人同士の特徴ベクトルの多数の組み合わせについてコサイン類似度を計算する。得られたコサイン類似度から正規化したヒストグラムを作成し、システムの他人受入率が 3.1 節で目標値として設定した 0.1%未満となるように閾値を決める。

3.3 消費電力を考慮した顔認証システムにおけるパラメータの設定

次のようなシステムを設計し、要素技術の評価によってバックボーン、各パラ

メータを決定する。設定するパラメータの一覧を表 2 に示す。

①バックボーンの設定

ResNet18、ResNet50、MobileNet の中で消費電力が少なく認証時間が短いものを選定する。

②検出後のスリープ時間の設定

検出回数が多くなると、認証を行わなくても消費電力が多くなってしまう。そこで検出を n 回行い、間に α 秒のスリープ時間を挟み認証の回数を減らす。また、全く顔が検出されなかった場合は長めに β 秒のスリープ時間を設定することで、消費電力を抑える。

③顔認証の際の閾値の設定

顔が検出された場合認証を行う。登録済みの顔画像の特徴ベクトルと、カメラで取得した入室者の顔画像の特徴ベクトルとの間でコサイン類似度を計算する。そして、その中で最も高い値を記録し、 γ 回連続でコサイン類似度が δ % を超えた場合を認証成功とする。この時、他人受入率が 0.1% 未満となるように値を設定する。

④顔検出頻度の設定

認証がうまくいかない場合に、すぐに β 秒の長いスリープに入ってしまうと入室者はその間は待つ必要があり、認証時間が増加する。それを防ぐために、一度でも顔が検出された場合はその後 ε 回の検出を行い、顔が ε 回連続で検出されなかった場合に β 秒のスリープを行う。

表 2 各パラメータの一覧

パラメータ	内容
n	1 度に行う検出の回数
α	検出後のスリープ時間 (秒)
β	n 回の検出で顔が検出できなかった際の長めのスリープ時間(秒)
γ	認証の際に連続で閾値を超える必要のある回数
δ	顔認証の際の閾値
ε	顔検出の頻度(連続回数)

4.要素技術の評価

4.1 バックボーンを変えた ArcFace による学習結果

VGGFace2 を用いて、ArcFace を学習した。学習時の設定を表 3 に示す。検証データには学習データは含まれていない。

表 3 学習時の設定

バッチサイズ	32
学習率(その他パラメータ)	$lr=0.01(\beta=0.9)$
最適化手法	SGD
エポック数	最大 100
学習データ	VGGFace2 より 8,847 人の画像 計 2,722,974 枚
検証データ	VGGFace2 より 8,847 人の画像 計 485,482 枚

表 4 学習結果

モデル	学習データ		検証データ	
	損失値	正解率	損失値	正解率
ResNet18	0.010180	0.947182	0.004679	0.916885
ResNet50	0.009273	0.954373	0.003787	0.936799
MobileNet	0.019200	0.897245	0.004002	0.929919

表 4 に学習結果を示す。学習の際は、損失関数の値がほとんど変化しなくなったところで打ち切った。損失値は学習終了時の損失関数(式(1))の値の事である。損失関数は、モデルが予測した結果と真のターゲット（目標）との差を定量化する指標であり、この差が小さいほど識別が正しく行われる。また、正解率はモデルが出した予測値が正解と一致している割合を表す。認証の精度にかかわる検証データの正解率に関しては、モデル間でほぼ同程度となった。

4.2 顔検出の精度評価と消費電力測定の結果

顔検出に OpenCV Haar-Cascade を用いて 5 分間検出のみを行い続け、その時の消費電力を測定した。その際、カメラの前に常に顔がある状態で行い、検出回数と顔の検出に成功した回数をカウントすることで、顔検出の精度を求める。カメラに顔を映し続けるためほぼ 100%に近い顔検出精度が期待される。また同時に、Raspberry Pi 起動のみの消費電力を測定して比較することで検出 1 回あたりにかかる消費電力を求める。結果を表 5 と図 4 で示す。

表 5 顔検出の結果

	総電流値 (mA)	平均電流値 (mA/s)	検出	検出 (成功)
起動のみ	106025	356.3133		
検出	214700	715.6667	8981	8969

表5の「総電流値」は5分間の毎秒の電流値の和、「平均電流値」は5分間の電流値の平均値を示している。また、「検出」はカメラから入力された画像内に顔があるかを判定した回数であり、「検出(成功)」は実際に顔があると判定された回数を示している。表5の検出を見ると OpenCV Haar-Cascade を用いた場合、Raspberry Pi カメラモジュールもフレームレートと同等の1秒当たり約30回の検出を行っていることが分かる。

顔検出精度は、次式で求めた。

$$\frac{\text{「検出(成功)」}}{\text{「検出」}} = 99.86\%$$

よって、カメラ内に顔がある場合はほとんど100%の確率で顔検出が可能であることが分かった。図4は起動のみと検出の電流値の推移を表している。横軸が時間(s)、縦軸が電流値(mA)となっている。検出によって消費電力も多くなっていることが分かる。本研究のシステムでは、3.1節で述べたように平均待ち時間の目標値は6秒であり、ここまで多く顔検出する必要はない。検出間に適切なスリープ時間を設定して処理を停止することで消費電力の削減につながる。

検出を1回行う場合に必要な電流値は次式で求めた。

検出の「総電流値」 - 起動のみの「総電流値」 = 12.10mA

「検出」

これにより、顔検出を行うと1回あたり 12.1mA の電流を消費することが分かった。

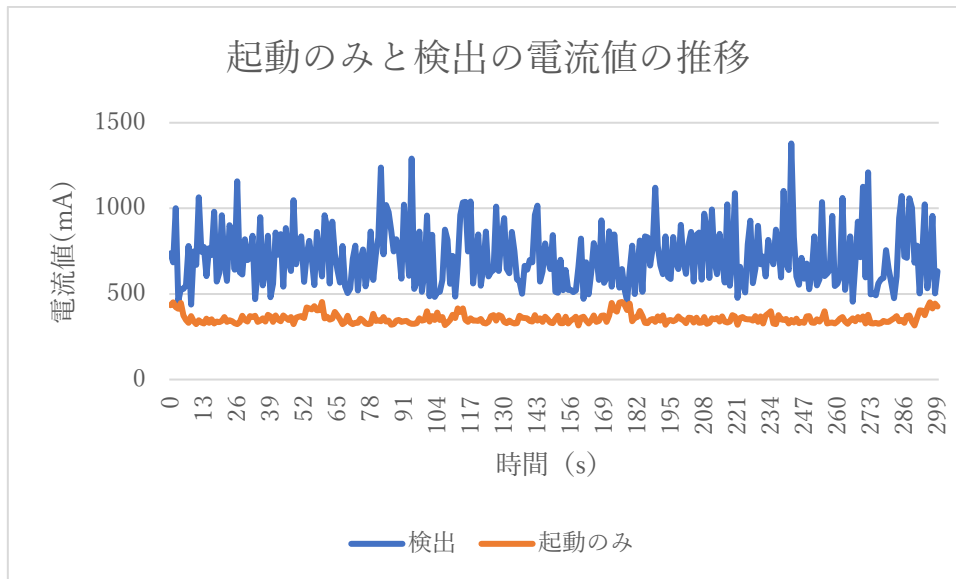


図 4 検出と起動のみの電流値推移

4.3 バックボーンの設定

学習した3つのモデル (ResNet18, ResNet50, MobileNet) を用いて顔認証を行う。その際に、RPZ-PowerMGRにより電流値を測定する。その結果を踏まえて、本研究の顔認証システムで用いるバックボーンを決定する。顔認証では、ArcFaceのバックボーンを変更した顔認証用のプログラムをRaspberry Pi上で実行させて、カメラに常に顔が映った状態で5分間認証を行い消費電力を測定した。認証では閾値は設けず、登録されている5つの顔画像の特徴ベクトルとコサイン類似度により比較を行う。なお、顔検出回数は4.2節の結果をふまえて、

検出間に 0.2 秒のスリープ時間を入れることで 1 秒当たり 5 回とし、検出に成功した場合に認証を行っている。ResNet18 の特徴ベクトルは 512 次元、ResNet50 は 2048 次元、MobileNet は 1024 次元となっている。

表 6 モデルごとの消費電力

モデル	総電流値 (mA)	平均電流値 (mA/s)	検出 (成功,認証)	検出 (失敗)	認証 1 回あたりの消費電力 (mA/s)
ResNet18	207,152	690.50	217	49	451.19
ResNet50	222,356	741.87	120	19	955.41
MobileNet	167,241	557.47	514	11	106.74

表 6 にモデルごとの消費電力を示す。表中の総電流値は 5 分間の毎秒の電流値の和を表し、平均電流値では 5 分間の電流値の平均値を表している。また、検出(成功,認証)は顔を検出し認証を行った回数であり、検出(失敗)は顔が検出できずに認証を行えなかった回数を表している。顔認証 1 回あたりの消費電力は次の式で求めた。

各モデルの「総電流値」－起動のみの「総電流値」－検出回数×検出 1 回に必要な消費電力

各モデルの「検出(成功,認証)」

表 6 の認証一回当たりの消費電力より、MobileNet を用いることで認証 1 回あ

たりにかかる電力を ResNet18 の約 4 分の 1、ResNet50 の 9 分の 1 まで減らせることが分かった。

次に、別途 50 回の認証を行い、モデルごとの平均認証時間と平均モデル処理速度を求めた。表 7 に結果を示す。ここでの平均認証時間とは検出を含めた認証時間のことを指す。一方で、平均モデル処理時間とは、検出後に画像を与えられたモデルが特徴ベクトルを抽出して登録されている画像の特徴ベクトルとコサイン類似度により認証を終えるまでの時間のことである。

表 7 モデルごとの平均認証時間

バックボーン	平均認証時間(s)	平均モデル処理時間(s)	検出(失敗)
ResNet18	1.396985	0.9204591	19
ResNet50	2.336721	1.8759600	0
MobileNet	0.561817	0.1071981	0

表 7 から、MobileNet が他の 2 つのモデルと比べて最も処理が早いことが分かる。消費電力、処理速度の観点から、本研究では ArcFace のバックボーンには MobileNet を使用する。

4.4 顔認証の際の閾値の決定

バックボーンに MobileNet を用いた ArcFace を学習した後、学習済のバックボーンを用いて、VGGFace2 の検証用の画像の特徴ベクトルを求めた。得られた特徴ベクトルを使って、本人同士と他人同士の特徴ベクトルの累計 1600 万通

りの組み合わせについてコサイン類似度を計算した。得られたコサイン類似度の分布(和が1になるように正規化したヒストグラム)を図6に示す。横軸がコサイン類似度の値、縦軸が相対頻度で、ビンの幅は、0.01としている。オレンジ色が他人同士の分布、青色が本人同士の分布を表す。これにより認証の際の閾値を決定する。

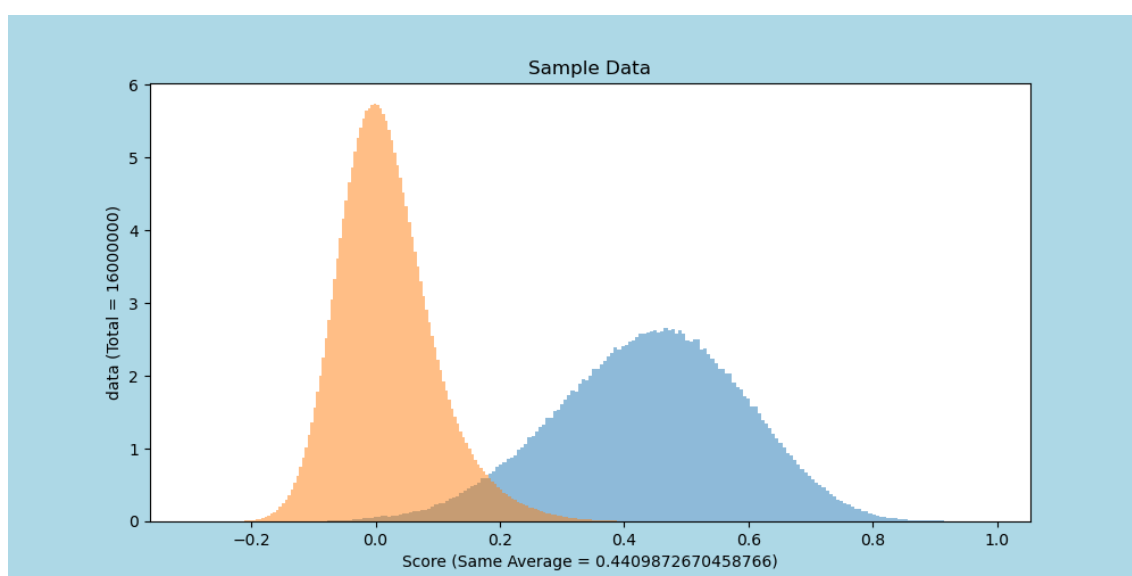


図 6 バックボーンに MobileNet を用いた ArcFace によるコサイン類似度

グラフが重なっている 0.2 付近で閾値を検討する。閾値を 0.21 に設定すると、他人同士を誤って同一人物と判定する他人受入率は約 2.94%となる。一方、本人同士を誤って他人と判定する本人拒否率は約 4.60%となる。3.1 節で定めた他人受入率の目標値は 0.1%であるので、閾値を 0.21 として、2 回連続で閾値を超えた場合に本人同士と認証することを考える。この場合、入室判定の他人受入率は

$$\text{入室判定の他人受入率} = \text{他人受入率} \times \text{他人受入率} = 0.0864\%$$

となり、目標値の 0.1%を切る。この時入室判定の本人拒否率は

$$\text{本人受入率} = 1 - \text{本人拒否率} = 95.4\%$$

$$\text{入室判定の本人拒否率} = 1 - \text{本人拒否率} \times \text{本人拒否率} = 0.0878\%$$

となる。

1 回の顔認証で入室判定をする場合、閾値を 0.37 とすると他人受入率が 0.0890%となり、0.1%を下回る。しかし、この時の本人拒否率は 27.1%と高い値になる。そのため、閾値としては適していない。

5. 設計した顔認証システム

本研究では顔認証システムにおける、バックボーン、パラメータの値、その他要素を、次のように設計した。設定したパラメータの値を表8に示す。

- ① MobileNet をバックボーンにした ArcFace を用いる
- ② 検出の間に $\alpha = 0.2$ 秒のスリープを入れ $n=7$ 回検出を行う。顔が検出されなかった場合に $\beta = 8$ 秒間のスリープを入れ、その後 $n=7$ 回検出を行う
- ③ 顔が検出された場合認証を行う。 $\gamma = 2$ 回連続でコサイン類似度が $\delta = 0.21$ を超えた場合を認証成功とする
- ④ 顔が一度でも検出された場合はそこから検出を $\varepsilon = 25$ 回行い、顔が 25 回連続で検出されなかった場合に②に戻る

表 8 各パラメータの設定値

パラメータ	内容
n	1 度に行う検出の回数 $n=7$ (回)
α	検出後のスリープ時間 (秒) $\alpha=0.2$ (秒)
β	n 回の検出で顔が検出できなかった際の長めのスリープ時間(秒) $\beta=8$ (秒)
γ	認証の際に連続で閾値を超える必要のある回数 $\gamma=2$ (回)
δ	顔認証の際の閾値 $\delta=0.21$
ε	顔検出の頻度(連続回数) $\varepsilon=25$ (回)

4章の要素技術の評価を受けて、組み込み機器上で動作する深層学習を用いた低消費電力顔認証システムとして ArcFace のバックボーンには ResNet18 や ResNet50 よりも MobileNet を用いることが適しているといえる。モデルの処理時間や消費電力も MobileNet は他に比べて格段に速く、少ない電力で認証が可能になる。また、閾値に関しては 4.4 節の結果に基づいて $\delta = 0.21$ とした。連続 $\gamma = 2$ 回の認証で入室判定をすることで、目標値の 0.1%未満の他人受入率を実現した。

顔認証システムに関して、初めにカメラから画像を読み込み顔検出を行うが、スリープ時間を挟みプログラムを一時的に停止させなければ、検出の増加でかえって消費電力が増えてしまうという問題があった。そこで解決策として、検出回数を $n = 7$ 回、検出間のスリープ時間を $\alpha = 0.2$ 秒とした。また、本研究で設定した状況では、認証しない時間の方が長い。そのため $n=7$ 回の検出で顔が検出できなかった場合はスリープ時間を 8 秒入れることで、消費電力を大幅に削減した。

次に、平均認証待ち時間を考える。認証を行い閾値を超えた場合を○、閾値に満たなかった場合を×で示す。 $\gamma = 2$ 回連続で閾値を超えた場合を入室許可とする。入室許可となる 3 通りのパターンについて、それぞれが生じる割合と、それぞれのパターンでの平均待ち時間を表 9 に示す。ここで、本人受入率は 4.4 節より 95.4%、MobileNet の平均認証時間は表 7 より 0.561817 秒とし、スリープ時間 $\beta = 8$ 秒の半分の 4 秒の待ち時間が発生するとしている。表 9 より、これらの

3パターンで99%以上の割合となる。この時の入室許可までの平均待ち時間は、5.374秒となる。これにより、 $n=7, \alpha=0.2, \beta=8$ とすることで、消費電力を削減しながら目標値であった6秒を切ることができた。

また、利便性の向上のために $\varepsilon=25$ として、顔検出後の5秒間は常に検出を行うことで認証失敗による認証時間の増加を防ぐ。

表 9 MobileNet を用いた場合の平均待ち時間

認証許可のパターン	発生確率(%)	平均待ち時間(s)
○ ○	91.012	5.324
× ○ ○	4.187	6.086
○ × ○ ○	3.994	6.848
	計 99.134	期待値 5.374

次に提案システムを5分稼働し、入室者2名の認証を行った場合の消費電力を測定した。なお、システムには5名分各1枚の顔画像が登録されており、この2名を含んでいるものとする。結果を表10及び図7に示す。

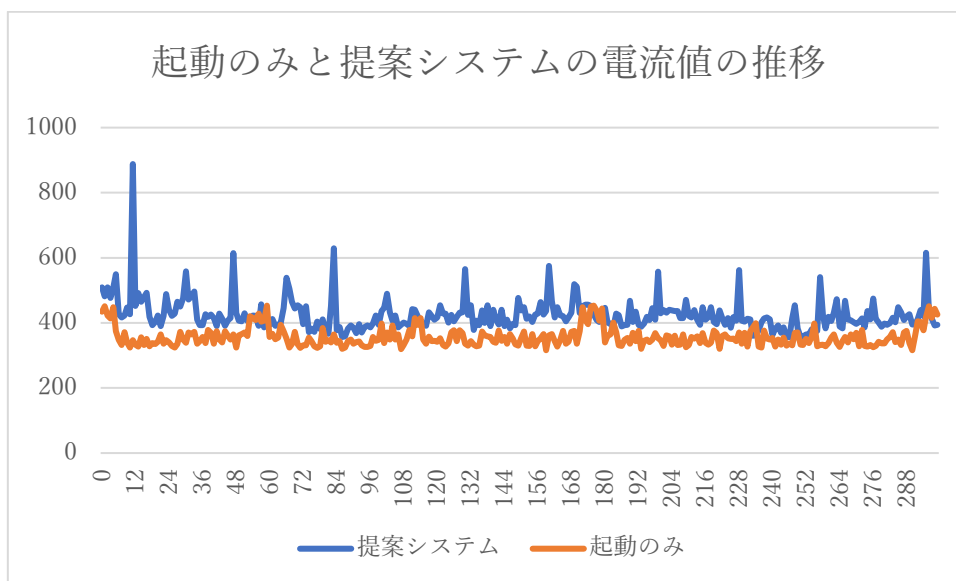


図 7 起動のみと提案システムの電流値の推移

表 10 起動のみと提案システムの電流値

	総電流値 (mA)	平均電流値 (mA/s)	検出 (成功)	検出 (失敗)
起動のみ	106025	356.31		
提案システム	126617	424.88	5	264

図 7 と表 10 から提案システムの消費電力が起動のみの値に近づいていることが分かる。また、目標値であった消費電力を起動のみの 120%以下にすることも達成できた。

6.結論

本研究では、顔認証システムに ArcFace を用いてそのバックボーンを変えながら電流値を測定し、その他のパラメータを決定することで、組み込み機器上で動作する低消費電力顔認証システムを提案した。結果として、バックボーンには MobileNet を用いた場合、ほぼ精度が変わらずに一回の認証当たり ResNet18 の約 4 分の 1、ResNet50 の約 9 分の 1 まで消費電力を抑えることができた。また、システム全体としての消費電力の削減を目指し、検出間にスリープ時間を設けることで消費電力を抑えながら、認証回数や閾値を適切に設定することで実用性と両立させた。

本研究で組み込み機器として使用した Raspberry Pi は性能が高く、起動のみでも多くの電力を消費していた。そのため、もっと基礎的な消費電力が少ない、小さな組み込み機器を用いた場合について検証することが、今後の課題として挙げられる。そのほか、本研究では VGGFace2 によるテストデータを用いて閾値を 0.21 と決定した。しかし、実用上閾値を 0.21 とすることは適正であるか定かではなく、学習データの偏りによっては閾値をもっと上げる必要性も考えられる。想定した状況に即した認証実験も今後の課題である。

謝辞

本研究を行うにあたり、多大な助言と支援をいただいた椋木雅之教授に感謝いたします。そして、研究の相談や議論を通して、多くの知識や示唆をいただいた椋木研究室の皆様にお礼申し上げます。

参考文献

- [1]business leaders square wisdom, “生体認証(バイオメトリクス認証)とは?”, <https://wisdom.nec.com/ja/feature/biometrics/2020032301/index.html> (参照 2024-1-25)
- [2]西崎博光, レオ チー シャン, 牧野浩二, “小型コンピュータにおける深層学習アプリケーションの動作検証”,電気学会論文誌 C,vol.138, pp.1108-1115, 2018
- [3] Jiankang Deng, Jia Guo, Niannan Xue, Stefanos Zafeiriou, “ArcFace :Additive Angular Margin Loss for Deep Face Recognition”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4690-4699, 2019
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep residual learning for image recognition”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016
- [5] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, “MobileNets: Efficient convolutional neural networks for mobile vision applications”, arXiv:1704.04861, 2017
- [6] Xinzheng Xu, Meng Du, Huanxiu Guo, Jianying Chang, Xiaoyang Zhao, “Lightweight facenet based on mobilenet”, International Journal of Intelligence Science, vol. 11, no. 1, pp. 1–16, 2020
- [7] T. V. Dang, “Smart home Management System with Face Recognition based on ArcFace model in Deep Convolutional Neural Network”, Journal of Robotics and Control, vol. 3, no. 6, pp. 754-761, 2022
- [8]Fushuai Wang, Renren Zheng, Penghui Li, Hanni Song, Dongming Du, Jingchao Sun. “Face recognition on Raspberry Pi based on MobileNetV2”, International Symposium on Artificial Intelligence and its Application on Media (ISAIAM), pp.116–120, 2021
- [9]Raspberry Pi,” Raspberry Pi”, <https://www.raspberrypi.com/> (参照 2024-2-3)
- [10] Zankruti Arya1, Vibha Tiwari, “Automatic Face Recognition and Detection Using OpenCV, Haar Cascade and Recognizer for Frontal Face”, International Journal of Engineering Research and Applications, vol.10, pp.13–19 ,2020
- [11] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, Andrew Zisserman, “VGGFace2: A dataset for recognizing faces across pose and age”, IEEE International Conference on Automatic Face & Gesture Recognition (FG), pp. 67–74, 2018