卒 業 論 文

DCT 係数上の高速な画像変換ライブラリの拡張

2023 年度

九州工業大学情報工学部

知能情報工学科

202C1071

島本 紘希

指導教員：坂本 比呂志　教授

		指導教員	坂本 比呂志　教授
学生番号	202C1071	氏　名	島本 紘希
論文題目	DCT 係数上の高速な画像変換ライブラリの拡張		

1　はじめに

人工知能の画像認識において, モデルの汎化能力を向上させることは重要である. その手法として, 学習データセットを拡張するデータオーギュメンテーションというものがある. これは画像を前処理をし, 学習に必要な画像数を補うというものである. また, 圧縮データの展開と再圧縮を避け, 直接変換を行う手法がある [1]. この処理を第三者サーバ上で行う場合, 画像データをサーバに送信することが必要となり, 機密情報漏洩が懸念される. この懸念点を改善するための手法 [2] が提案されたが, 拡大・縮小に関しては実装が成されていなかった. そのため, 本研究では, 圧縮画像データを直接拡大・縮小する手法を提案する.

2　提案手法

関連研究 [3] の手法を応用し, DCT 変換された画像を任意の倍率で拡大・縮小する手法を提案する.

2.1　拡大

$M \times N$ 行列 A を, a 倍 $(a > 1)$ し, $aM \times aN$ 行列 A_a にする式を式 (1) に示す.

$$A_a = \begin{bmatrix} \alpha I_8 \\ 0 \end{bmatrix}_{aM \times M} A_{M \times N} \begin{bmatrix} \alpha I_8 & 0 \end{bmatrix}_{N \times aN} \quad (1)$$

ただし, $\alpha = 1.5 + 0.3 \times (a - 2)$

2.2　縮小

$M \times N$ 行列 A を, a 倍 $(0 < a < 1)$ し, $aM \times aN$ 行列 A_a にする式を式 (2) に示す.

$$A_a = \begin{bmatrix} \alpha I_8 & 0 \end{bmatrix}_{aM \times M} A_{M \times N} \begin{bmatrix} \alpha I_8 \\ 0 \end{bmatrix}_{N \times aN} \quad (2)$$

ただし, $\alpha = a + (1 - a)/2$

3　実験結果

これから示す縦棒グラフの青色の縦棒が提案手法, オレンジ色の縦棒が従来手法の処理時間である. また, 誤差範囲は標準偏差を表す.

3.1　拡大の実験結果

倍率が 2 倍, 3 倍, 4 倍, 5 倍の場合に提案手法と従来手法の処理時間を比べた結果は図 1 の通りである.

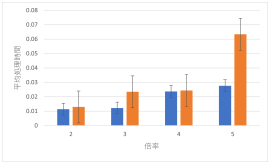

図 1: 拡大へ近処理時間の比較

すべての倍率において, 提案手法が従来手法より処理時間が短いことがわかった.

3.2　縮小の実験結果

倍率が 1/8 倍, 1/4 倍, 1/2 倍, 3/4 倍の場合に提案手法と従来手法の処理時間を比べた結果は図 2 の通りである.

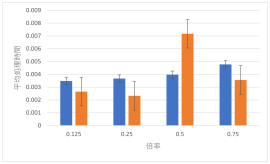

図 2: 縮小平均処理時間の比較

倍率が 0.5 のときのみ従来手法より平均処理時間が短いという結果が得られた.

3.3　拡大の秘匿計算の結果

提案手法の処理時間は, 188.744[s], 従来手法の処理時間は, 617.933[s] となった. また, 復号された行列は, 4×4 行列から 8×8 行列になっているが, 絶対値が 255 を大きく超える値が出現する結果となった.

4　まとめ

拡大は提案手法が従来手法より処理時間が短いという結果が得られたが, 縮小は提案手法が従来手法より処理時間が長いという結果となった.

参考文献

[1] B. Shen and I. K. Sethi, “Inner-Block Operations On Compressed Images” Proc. ACM Intl. Conf. Multimedia ’95, pp. 490-499 San Francisco, Nov. 1995.

[2] 日比 貫智, ”準同型暗号による DCT 係数上の高速な画像変換” 九州工業大学修士論文, 2022

[3] H. Shu, et al. “An efficient arbitrary downsizing algorithm for video transcoding” IEEE Trans. Circuits Syst. Video Technol., 2004, 14, (6), pp. 887–891.

目 次

第1章　はじめに

人工知能の画像認識の分野において, 深層学習は優れた能力を発揮する. しかし, モデルの汎化性能を向上させることは, 数ある問題の中でも困難な課題の一つである. 汎化性能の向上させる手法として、学習データセットを拡張するデータオーギュメンテーションというものがある. 深層学習モデルの学習には十分に大きな画像数が必要であり, データオーギュメンテーションによって, 画像に前処理を行い, 画像数不足を補うという手法が一般的である. データオーギュメンテーションには、画像の回転, 反転, 切り取り, 明度・コントラスト調整, ノイズの追加, 拡大・縮小などのような手法が挙げられる. データオーギュメンテーションの一例を図 1.1 に示す. また, 画像変換は圧縮データから展開を行い, 変換を行ってから, 再圧縮という流れが一般的であったが, 圧縮データの展開と再圧縮を避け, 圧縮データに対して直接変換を行う手法がある [1]. この処理を第三者のサーバ上で行う場合, 画像データをサーバに送信することが必要となり, 第三者のサーバから機密情報漏洩が懸念される. この懸念点を改善するための手法 [2] が提案されたが, 拡大・縮小に関しては実装が成されていなかった. そのため, 本研究では, 離散コサイン変換された画像データを直接拡大・縮小する手法を提案する.

元画像　　左右反転　　上下反転

図 1.1 データオーギュメンテーションの例

第2章　前提知識

2.1　JPEG 圧縮

JPEG 圧縮とは, 一般的に用いられる画像データの非可逆圧縮方式の一つである. カラー画像の場合, RGB 色空間の画素を YCbCr 色空間の画素へ変換し, クロマサブサンプリングを行う. 画像を 8×8 画素のブロック単位で分割 (ブロック分割) し, そのブロック単位で離散コサイン変換 (DCT) を用いて, 色空間領域から周波数領域へと変換する. この処理で DCT 係数へと変換された画像を量子化テーブルを用いて量子化する. このように量子化された DCT 係数は, 高周波成分が 0 になりやすい特性を持つ. 0 が多いという特性を利用し, ゼロランレングス圧縮を行って, ハフマン符号を用いたエントロピー符号化により圧縮が行われる. この節では, JPEG 圧縮の中でも, 本研究に関連するブロック分割と離散コサイン変換を説明する.

2.1.1　ブロック分割

図 2.1 のように, 画像を 8×8 画素のブロック単位に分割する. 画像の高さや幅が 8 で割り切れない場合は 8 で割り切れるようにパディングを行う. 図 2.1 は 256×256 の画像であるため, 縦横 32 分割されている.

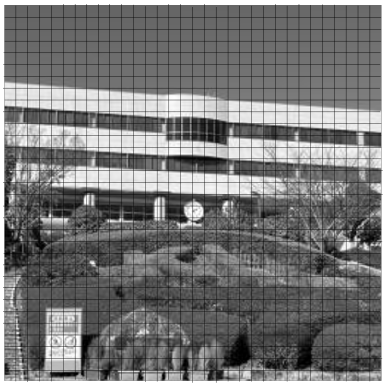

図 2.1 ブロック分割の例

2.1.2 離散コサイン変換 (DCT)

ブロック分割された各ブロックに対して離散コサイン変換を行い, ブロックの表現を空間領域から周波数領域へと変換する. $M \times N$ 行列 A の二次元離散コサイン変換は式 (2.1) で定義される. このとき, $B_{p,q}$ は A の DCT 係数といい, DCT 係数 B から元の行列 A を復元する場合は, 式 (2.2) によって定義される逆離散コサイン変換 (IDCT) を行う.

$$B_{p,q} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{m,n} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N} \tag{2.1}$$

$$A_{p,q} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \alpha_p \alpha_q B_{m,n} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N} \tag{2.2}$$

α_p, α_q は以下のように定義される.

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}} & (p = 0) \\ \sqrt{\frac{2}{M}} & (1 \leq p \leq M-1) \end{cases}$$

$$\alpha_q = \begin{cases} \frac{1}{\sqrt{N}} & (p = 0) \\ \sqrt{\frac{2}{N}} & (1 \leq p \leq N-1) \end{cases}$$

2.2 データオーギュメンテーション

深層学習を用いた画像処理を行う場合, 学習に大量のデータを要する. この問題に対処する方法の一つが, 「学習するデータを前処理し, オリジナル画像にいくつかの基本的な画像処理をランダムに適用して新たな画像を生成する」である. この画像処理のことをデータオーギュメンテーションという. データオーギュメンテーションは, 深層学習の過学習やモデルの汎化性能を向上させることにも貢献できる非常に重要な分野である. データオーギュメンテーションには様々な手法が存在する. 本研究では, データオーギュメンテーション手法のうち, 拡大・縮小を行う.

2.3 準同型暗号

暗号化したデータを復号することなく, 暗号化したまま処理を行うことができる暗号方式を, 準同型暗号という.

2.3.1 完全準同型暗号

準同型暗号の中でも, 任意の回数の加算と乗算を行うことができるものを完全準同型暗号と呼ぶ. すなわち, Enc を暗号化関数, $\cdot$ を二項演算とすると, 2 つの平文 x, y の二項演算 $x \cdot y$ を, 暗号化された $Enc(x)$ と $Enc(y)$ を復号することなく $Enc(x \cdot y)$ で計算することができる.

2.3.2 TFHE

完全準同型暗号では，平文にランダムなノイズを加えて暗号化することで情報を秘匿化する．このノイズは準同型演算を行うたびに増加し，ノイズが大きくなるほど暗号文を正しく復号できない確率が高くなる．完全準同型暗号では，暗号文に Bootstraping とよばれる処理を施すことで，暗号文のノイズを削減することができ，これにより任意の回数の準同型演算を実行しても，暗号文を正しく復号することができる．一般に，Bootstrapping の処理は低速であるが，完全準同型暗号のなかでも [4] の Bootstrapping は比較的高速である.

2.4 Arbitrary Downsizing Algorithm

関連研究 [3] で案された手法であり, 任意のサイズ ($R_x \times R_y$, ただし R_x は行列の行数, R_y は行列の列数) の離散コサイン変換された画像に対して, 高周波成分のほとんどを破棄し, 低周波成分のみを抽出して, サイズが 8×8 のブロックとする手法である. 自然画像において、DCT 係数上では信号の大部分のエネルギーが低周波数領域に集中している. そのため, ブロックの高周波成分を破棄し, 低周波成分のみを保持することで元々のブロックのほとんどのエネルギーを保持することができる. 提案された手法を式 (2.3) に示す.

$$\hat{B} = \begin{bmatrix} I_8 & 0 \end{bmatrix}_{8 \times 8R_x} B \begin{bmatrix} I_8 \\ 0 \end{bmatrix}_{8 \times 8R_y} \tag{2.3}$$

このとき, I_8 は 8×8 の単位行列である. また, 行列 B は, 離散コサイン変換された $R_x \times R_y$ 行列 A を, $8R_x \times R_x$ の行列 M_{ijL} 及び, $R_y \times 8R_y$ の行列 N_{ijR} によって,

$$\bar{A} = M_{ijL} A N_{ijR}$$

$$B = \sum_{i=1}^{M} \sum_{j=1}^{N} \bar{A_{ij}}$$

とした行列である. これは, 画像の任意の箇所を切り取る操作である. 本研究では, 画像の任意の範囲ではなく, 画像全体を拡大・縮小する方法を提案する.

第3章　提案手法

本研究では, 関連研究 [3] にて提案された, 任意のサイズ任意のサイズ $(R_x \times R_y)$ から, 8×8 に縮小するという方法を応用し, DCT 変換された画像を任意の倍率で拡大・縮小する手法を提案する. 実験に使用した画像は図 3.1 の 256×256 の画像である. また, 式 (3.1), 式 (3.2) にそれぞれ定義される α は, 変換した画像が元画像より暗くなったり, 明るくなったりしたため, 複数個の倍率で実験を繰り返し元画像に近くなるように倍率ごとに変化する定数である. 提案手法の倍率は実数に対応しているが, 小数点以下は切り捨てる処理を行っている.

図 3.1 256×256 の画像

3.1　拡大

$M \times N$ 行列 A を, a 倍 $(a > 1)$ し, $aM \times aN$ 行列 A_a にする式を式 (3.1) に示す.

$$A_a = \begin{bmatrix} \alpha I_M \\ 0 \end{bmatrix}_{aM \times M} A \begin{bmatrix} \alpha I_N & 0 \end{bmatrix}_{N \times aN} \tag{3.1}$$

このとき, α は以下のように定義される.

$$\alpha = 1.5 + 0.3 \times (a - 2)$$

3.2 縮小

$M \times N$ 行列 A を, a 倍 $(0 < a < 1)$ し, $aM \times aN$ 行列 A_a にする式を式 (3.2) に示す.

$$A_a = \begin{bmatrix} \alpha I_{aM} & 0 \end{bmatrix}_{aM \times M} A \begin{bmatrix} \alpha I_{aN} \\ 0 \end{bmatrix}_{N \times aN} \tag{3.2}$$

このとき, α は以下のように定義される.

$$\alpha = a + \frac{1-a}{2}$$

3.3 実験方法

本研究では, IDCT から拡大・縮小し DCT を行う従来手法と DCT 係数から直接変換する提案手法の処理時間を比較する実験を行う. 従来手法の拡大・縮小には, 一般的に使われる OpenCV の resize 関数を用いる. また, 本研究では, DCT 係数のまま画像の拡大・縮小を行うことを主目的とするが, 第 1 章に述べた, 処理を第三者サーバで行う場合の機密情報漏洩の懸念点がある. そのため, 付属実験として, TFHE を用いた拡大の秘匿計算も試みる. 実験方法は, 一つの成分が 0 255 の値で定められた 4×4 行列を暗号化し, 提案手法と従来手法で処理時間を比較し, 復号化する実験を行う.

第4章　実験結果

これから示す縦棒グラフの青色の縦棒が提案手法, オレンジ色の縦棒が従来手法の処理時間である. また, 誤差範囲は標準偏差を表す.

4.1　拡大

図 4.2, 図 4.3 はそれぞれ拡大前と拡大後の画像である. また, 図 4.1 は実行結果の例である.

```
(256, 256)
100
Proposed method
0.004905939102172851 6
(512, 512)
112.92172
Conventional method
0.006895303726196289
True
```

図 4.1 拡大実行結果の例

ここで, (256, 256) は元画像の大きさで, (512, 512) は出力された画像の大きさである.

図 4.2 拡大前

図 4.3 拡大後

倍率が 2 倍, 3 倍, 4 倍, 5 倍の場合に従来手法と提案手法の処理時間を比べた結果は図 4.4 の通りである.

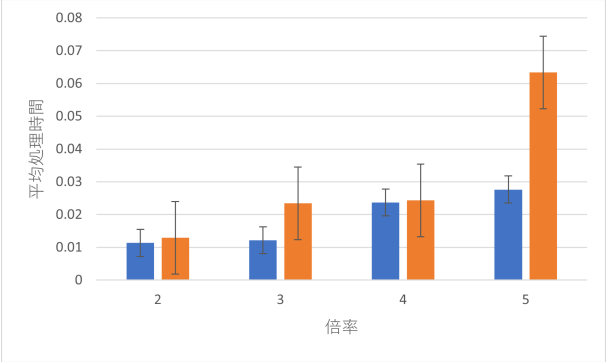

図 4.4 拡大平均処理時間の比較

従来手法より, 提案手法が平均処理時間が短いという結果が得られた.

4.2 縮小

図 4.6, 図 4.7 はそれぞれ縮小前と縮小後の画像である. また, 図 4.5 は実行結果の例である.

```
(256, 256)
Proposed method
0.0012617111206054688
(128, 128)
(256, 256)
(128, 128)
Conventional method
0.003017902374267578
True
```

図 4.5 縮小実行結果の例

ここで, (256, 256) は元画像の大きさで, (128, 128) は出力された画像の大きさである.

図 4.6 縮小前

図 4.7 縮小後

倍率が 0.125 倍, 0.25 倍, 0.5 倍, 0.75 倍の場合に従来手法と提案手法の処理時間を比べた結果は図 4.8 の通りである.

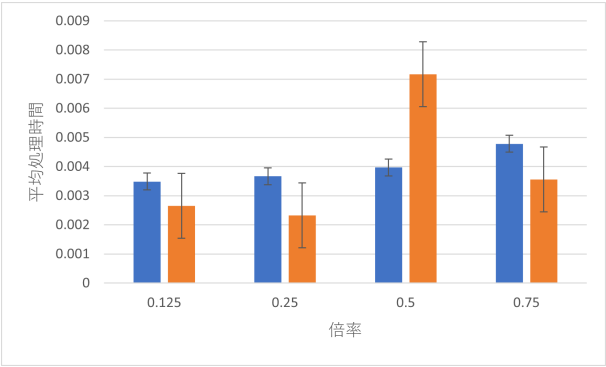

図 4.8 縮小平均処理処理時間の比較

倍率が 0.5 のときのみ従来手法より平均処理時間が短いという結果が得られた.

4.3 準同型暗号による拡大秘匿計算

実行結果の一例を示す.
ランダムに生成された行列が matrix1 である.
matrix1 =
84 65 9 67
251 8 44 44
35 150 152 197
82 241 55 158
提案手法の処理時間は, 188.744[s]
従来手法の処理時間は, 617.933[s]
となった.
さらに, 提案手法を復号した結果が result1, 従来手法を復号した結果が result2 である.
result1 =
-2136997888 1669857280 231211008 1721237504 0 0 0 0
-2141716480 205520896 1130364928 1130364928 0 0 0 0
899153920 -441450496 -390070272 765984768 0 0 0 0
2106589184 1896349696 1412956160 -235929600 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

result2 =
1073741831 0 -1879048192 0 268435456 0 -1342177277 0
0 0 0 0 0 0 0 0
805306368 0 -2147483644 0 -1073741824 0 -1073741824 0
0 0 0 0 0 0 0 0
-1342177279 0 1610612736 0 -2147483643 0 -805306368 0
0 0 0 0 0 0 0 0
536870914 0 -1879048191 0 -268435452 0 -536870908 0
0 0 0 0 0 0 0 0

処理時間を比較すると, 提案手法の処理時間が短いことが分かる. また, 復号された行列を確認すると, 4×4 行列から 8×8 行列になっているため, 拡大には成功しているように見える. しかし, それぞれ絶対値が 255 を大きく超える値が出現している結果となった.

第5章　考察

5.1　拡大・縮小の考察

拡大に関しては, 実験を行ったすべての倍率において, 提案手法が従来手法より平均処理時間が短いことが分かった. しかし, 縮小に関しては, 実験を行った倍率の中で 0.5 倍の場合でのみ, 提案手法が従来手法より平均処理時間が短いことが分かった. これにより, 行列を用いた拡大・縮小の方法では拡大の方がより優れた手法であるといえる. 提案手法の縮小平均実行時間が, 従来手法の縮小平均実行時間より長くなった理由は, 縮小という処理に行列の積という処理が適していないからではないかと考えられる. 一般的に行列の積のアルゴリズムは繰り返し処理を 3 重で行う. そのため, 処理時間が長くなりやすい特徴がある. これが IDCT, 縮小, DCT を行った従来手法の処理時間より, 長くなったことが原因と考えられる.

5.2　準同型暗号による拡大秘匿計算の考察

復号化した行列の成分の中に絶対値が 255 より大きな値が出現したのは, DCT, IDCT の計算, もしくは復号化に失敗しているからではないかと考えられる. 準同型暗号には, 計算回数が多くなるごとにノイズが大きくなり正しく復号されない場合がある. そのため, いずれかの計算過程でノイズが大きくなり, Bootstrapping の処理を通してもうまく復号されなかったとことが原因と考えられる.

第6章　まとめ

拡大は提案手法が従来手法より処理時間が短いという結果が得られたが, 縮小は提案手法が従来手法より処理時間が長いという結果となった. そのため, 今後は縮小の方法をより早く処理できる手法を考え, 準同型暗号による秘匿計算もあわせて実装していきたい.

謝辞

本研究を行うにあたって, 一年間ご指導くださった坂本先生, 研究室の先輩方, 同級生に感謝申し上げます.

参考文献

[1] B. Shen and I. K. Sethi, “Inner-Block Operations On Compressed Images” Proc. ACM Intl. Conf. Multimedia ’95, pp. 490-499 San Francisco, Nov. 1995.

[2] 日比 貫智, ”準同型暗号による DCT 係数上の高速な画像変換” 九州工業大学修士論文, 2022

[3] H. Shu, et al.“An efficient arbitrary downsizing algorithm for video transcoding” IEEE Trans. Circuits Syst. Video Technol., 2004, 14, (6), pp. 887–891.

[4] I. Chillotti, et al. “Tfhe: fast fully homomorphic encryption over the torus” Journal of Cryptology, Vol. 33, No. 1, pp. 34–91, 2020.