

飯國研ゼミ第11回 7/4(火)

画像処理によるフェイク画像の判定

基礎工学部システム科学科
知能システム学コース4年
09C20020 今井智也

テーマ：
画像処理によるフェイク画像や動画の判定



目次

- 1. 研究背景
- 2. 実装方法
- 3. 実験結果
- 4. 結論

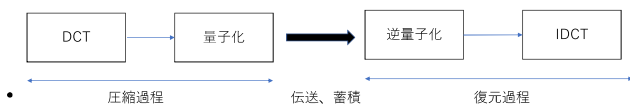
実装方法

- 改ざん部分のブロック境界の高周波成分が大きくなることを利用



JPEGのデータ圧縮

- JPEG画像は8×8画素をブロック単位で離散コサイン変換する
- それを量子化してデータを削減する（高周波成分ほど大きく量子化される）



離散コサイン変換

$$0 \leq u, v < T$$
$$F(u, v) = \frac{1}{T} C(u) C(v) \sum_{x=0}^{T-1} \sum_{y=0}^{T-1} f(x, y) \cos\left(\frac{(2x+1)u\pi}{2T}\right) \cos\left(\frac{(2y+1)v\pi}{2T}\right)$$
$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{else} \end{cases}$$

```
DCT weights
def w(x, y, u, v):
    cu = 1.
    cv = 1.
    if u <= 0:
        cu = np.sqrt(2)
    if v <= 0:
        cv = np.sqrt(2)
    theta = np.pi / (2 * T)
    return ((2 * cu * cv / T) * np.cos((2*x+1)*u*theta) * np.cos((2*y+1)*v*theta))

DCT
def dct(img):
    H, W, _ = img.shape
    F = np.zeros((H, W, channel), dtype=np.float32)
    for c in range(channel):
        for y1 in range(0, H, 8):
            for x1 in range(0, W, 8):
                for v in range(1):
                    for u in range(1):
                        for x in range(8):
                            for y in range(8):
                                F[y1+y, x1+x, c] += img[y1+y, x1+x, c] * w(x, y, u, v)
    return F
```

DCT成分解析

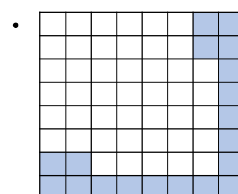
- 改ざん画像をJPEGで保存することで改ざん部のブロック境界により改ざん部のブロックの高周波成分が大きくなる
- DCT係数のうち高周波のDCT係数のみを抜き出す



- 8×8ブロックの取り方を全通り行い(左上の座標を(8×m+s, 8×n+t)(1≤s,t≤8)にずらした64通りのDCT係数を調べる)ブロックごとの総和Sを絶対値をとったうえで求め、S>閾値となる領域を探す

DCT成分解析

- 1 DCT係数の高周波成分を抜き出す



```
dct_coefficients = [
    (0, 7), (7, 0), (1, 7), (7, 1),
    (2, 7), (7, 2), (3, 7), (7, 3),
    (4, 7), (7, 4), (5, 7), (7, 5),
    (6, 7), (7, 6), (7, 7), (0, 6),
    (6, 0), (1, 6), (6, 1)
]
```

DCT成分解析（実験）

- 閾値以上の領域を黒く塗りつぶす



図1 元画像

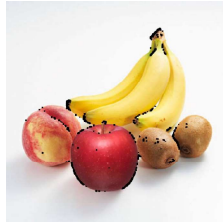


図2 実験結果 1 (閾値 60)

DCT成分解析（実験）

- 黒く塗りつぶした領域を保持する



図3 元画像



図4 実験結果 2 (閾値 60)

DCT成分解析（実験）

- 閾値以上の領域を黒く塗りつぶす



図5 元画像

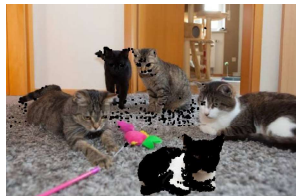


図6 実験結果 3 (閾値 100)

DCT成分解析（実験）

- 黒く塗りつぶした領域を保持する



図7 元画像

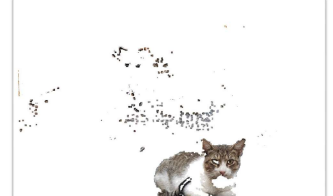


図8 実験結果 4 (閾値 100)

DCT成分解析（実験）

- 閾値以上の領域を黒く塗りつぶす



図9 元画像



図10 実験結果 5 (閾値 130)

DCT成分解析（実験）

- 黒く塗りつぶした領域を保持する



図11 元画像



図12 実験結果 6 (閾値 130)

DCT成分解析（実験）

- 閾値以上の領域を黒く塗りつぶす

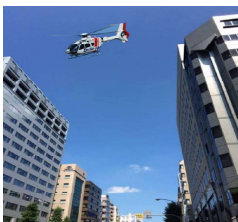


図13 元画像

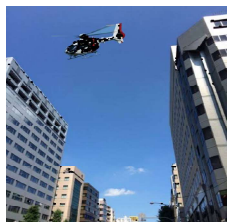


図14 実験結果 7 (閾値 150)

DCT成分解析（実験）

- 黒く塗りつぶした領域を保持する

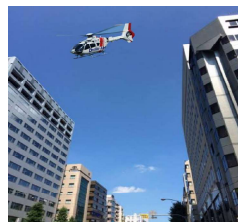


図15 元画像



図16 実験結果 8 (閾値 150)

DCT成分解析（実験）

- 閾値以上の領域を黒く塗りつぶす



図17 元画像



図18 実験結果18（閾値 140）

DCT成分解析（実験）

- 黒く塗りつぶした領域を保持する



図19 元画像

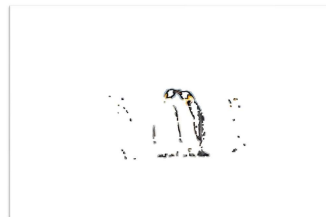


図20 実験結果20（閾値 140）

DCT成分解析（実験）（失敗例）

- 閾値以上の領域を黒く塗りつぶす



図21 元画像



図18 実験結果22（閾値 100）

DCT成分解析（実験）（失敗例）

- 黒く塗りつぶした領域を保持する



図23 元画像

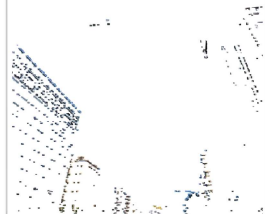


図24 実験結果10（閾値 100）

結論と今後の展望

- 画像によってうまくいくものとうまくいかないものがあった
- 各画像に対して適切な閾値の値が異なる
- 加工部分以外にも条件を満たす部分が多く存在する

参考文献

- <https://digitalforensic.jp/wp-content/uploads/2019/03/3-uehara.pdf>（デジタル画像改ざんに関する研究）
- <https://www.ieice.org/publications/conference-FIT-DVDs/FIT2015/data/pdf/I-006.pdf>（DCT成分解析等を用いたJPEG画像の整合的改ざん検出ツールの開発）
- <http://www.nilim.go.jp/lah/bcg/sinyou/tnn0361pdf/ks0361005.pdf>（国土技術政策総合研究所）
- <https://zenn.dev/youheinakagawa/articles/6d381d3a524367061f07>（画像加工検知に関するメモ）
- https://github.com/minido/Gasyori100knock-1/tree/master/Question_11_20（画像処理100本ノック）
- <https://www.lifewithpython.com/2014/12/python-extract-exif-data-like-data-from-images.html>（pythonからexif情報を取得したい）
- <https://arakan-pgm-ai.hatenablog.com/entry/2020/08/10/000000>（python3でExif情報の参照・削除・GPSから緯度経度計算）
- 美濃 導彦 画像処理論 Web情報理解のための基礎知識 第1版（2014）
- 大阪大学基礎工学部「信号処理」講義資料 2022 担当 飯岡洋二
- 奈良先端科学技術大学院大学 OpenCVプログラミング作成チーム OpenCV プログラミングブック 第2版（2010） 株式会社 毎日コミュニケーションズ