令和 4 年度
プロジェクトデザイン III
プロジェクトレポート
情報工学科

電車の分類をしたよ

提出日　令和 99 年 99 月 99 日

指導教員：鷹合 大輔 准教授

氏名　学籍番号 (クラス–名列番号)
野崎 悠度　1936463 (4EP1–68)
奥野 細道　1936119 (4EP4–75)

概要

本プロジェクトでは，機械学習を用いた電車の車両タイプを分類するシステムの開発を行った．
（目的）
世の中には似ているようなものでも，実は同じではないものがある．そのようなものを判断できるようになりたい．
（背景）
電車の車両タイプの種類は,JR の在来線だけでも１００種類近く存在している．多くの人は電車を見て，電車だと認識することは可能であるが，その電車の車両タイプまでを判断できる人は少ない．電車についての知識がある人は一目見るだけでその電車の車両タイプを判断できるが，大多数の人は似ている電車の車両タイプを判断することが難しい．そのため，だれでも簡単に電車の車両タイプを判断できるシステムの開発を行った．今回は JR 西日本の１７種類の電車を分類するものと，新幹線の各車両タイプを分類するものの二種類のシステム開発を行った．
（手法）
開発の手順は，まずは画像を集めてデータセットを作成する．次に作成したデータセットと YOLO を用いて学習させ評価を行った．学習データは車両タイプ別に YouTube にアップロードされている動画から電車が写っている場面を切り出した画像を利用した．一つの動画から一種類の車両タイプのデータセットを作成すると，似たような画像が大量に保存されてしまう．複数の動画の任意の場面を連結して一本の動画にするシステムも作成し，様々な場面の電車の画像が保存できるようにした．学習がこれ以上向上しないというところまで学習させ，モデルを作成した．作成したモデルを使ってブラウザ上で電車の車両タイプを分類する WEB アプリも作成した．
（結果）
特に似ている３種類の電車以外はだいたい分類することができた．似ている３種類は，誤分類が多かった
JR 北 (5) JR 東 (33) JR 東海 (11) JR 西 (25) JR 四国 (5) JR 九州 (15)

活動履歴 山田 太郎

期間	活動内容	活動時間 [h]
4–8 月	調査・実験・実装	200
11–3 月	実験・実装・論文執筆（3 章）	160

活動履歴 鬼 太郎

期間	活動内容	活動時間 [h]
4–8 月	調査・実験・実装	200
11–3 月	実験・実装・論文執筆（3 章）	160

活動履歴 目玉野 親父

期間	活動内容	活動時間 [h]
4–8 月	調査・実験・実装	200
11–3 月	実験・実装・論文執筆（3 章, **??**節）	160

目次

図目次

表目次

ソースリスト目次

第 1 章

序論

（背景）
現在，似ている電車がいっぱいある
（目的）
それぞれがどの車両タイプの電車なのかを判断したいよ画像を読み込ませることで，その画像には何が写っているのか判断するシステムを開発する
(問題点)
画像に写っている電車が何なのかを判断するためには電車の図鑑と画像を見比べて，自分で判断する方法しかない．その車両が何なのか知るために大きな労力が必要であることが問題である．
(現在の手法)
現在の画像分類では，画像に写っているのが，人や車，犬など，大まかな分類しかすることができない．この方法では電車の車両タイプが何かを判断することができない．

第 2 章

システム概要

2.1　この章で書くこと

- 識別，分類とは
- 現存するサービスでできること
- 概要図
- モデルについて
- サーバ関連について

2.2　分類・識別とは

分類とは，画像に写っているものが何かを判断することである．
(例) 犬，猫，電車，人
識別とは，画像に写っているものが何か，どこに写っているのかを判断することである．
(例) 家族の集合写真で，自分がどこにいるのかを特定する．

2.3　現存するサービス

- Google レンズ
- YOLO

2.3.1　Google レンズ

これは画像の分類を行うアプリである．単に分類結果が表示されるのではなく，分類したいオブジェクトが映っているウェブサイト一覧が表示されるものである．表示されたウェブサイトを適当に選び自分が知りたい結果をウェブサイトの中から探し出す必要がある．また，一枚の画像に複数のオブジェクトが存在している場合は正しい結果が得られない．

2.3.2　YOLO

YOLO の説明いいいいい
YOLO では，学習データを準備し学習させることで任意のオブジェクトを判別できるモデルの開発ができる．本プロジェクトでは，YOLO を用いて電車の車両タイプを識別，分類する 2 つのモデルを開発する．

2.4　システム概要図

本プロジェクトで開発するシステム概要を図 2.1 に示す．システムは車両判別部と UI に分けられる．
田村と相談，画像は後で変える

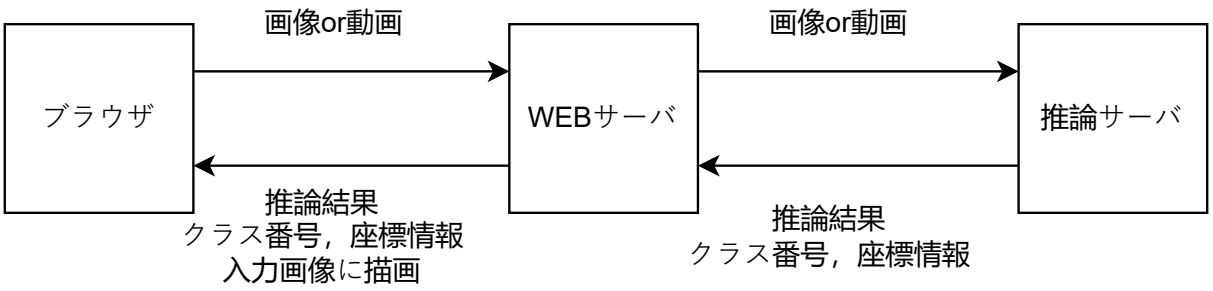

画像or動画
画像or動画
ブラウザ
WEBサーバ
推論サーバ
推論結果
クラス番号，座標情報
入力画像に描画
推論結果
クラス番号，座標情報

図 2.1: システム概要図

第 3 章

学習データの準備

3.1　この章で書くこと

- django の件
- 電車が映っている場面だけ画像で保存
- データセットを作成（識別・分類）
- アノテーションについて
-

3.2　画像の収集

学習データは，ある電車が映っている YouTube の動画をダウンロードして，任意の枚数だけランダムでフレームを保存する．その後，保存した画像を識別して電車が映っている場面のみ画像で保存をした．本プロジェクトでは，JR 西日本の在来線の電車を識別する．
205 207 213 221 223 225 227 271 281 283 285 287 321 323 381 521 683
上記の１７種類の画像を集める

3.2.1　動画の保存方法

動画の保存には yt-dlp というものを使う．１～３種類の動画の任意の秒数をダウンロードしてそれぞれの動画を連結して一つの動画にする web アプリを python のフレームワークの一つである django を使って作成した．
画像は後で変える，大きさを揃える
作成した web アプリは，図 3.1 図 3.2 図 3.3 のようになっている．YouTube 上の動画の URL と開始時刻 (秒), 終了時刻 (秒), 車両タイプ名を入力して，1～3 種類の動画を保存し連結して一つの動画にしている．

3.2.2　画像の保存方法

画像の保存は二回行う一回目は，保存した動画でランダムなフレームを保存する．二回目は，保存した画像を配布されているモデル (yolov8n.pt) で識別をして，電車が一つだけ映っているものだけを保存する．電車の画像を保存する際には，車両タイプ（数字３桁）というディレクトリに
車両タイプ（数字３桁）__通し番号（数字４桁）.jpg という名前で保存する

3.3　データセットの作成

3.3.1　データセットの構造

識別モデルと分類モデルでは学習時に使用するデータセットの構造がことなる．また識別モデル用のデータセットには画像のアノテーション情報が必要となる．

3.3.2　アノテーション

アノテーションとは、機械学習の分類の一つである教師あり学習において, 分析対象データにラベルを付与するプロセスである. 画像にバウンディングボックスと呼ばれる四角形を描画しクラス番号を指定する．バウンディングボックスを描画することでその画像に写っている物体の座標情報を取得することができる．クラス番号とは，判別したいものリストを作成し，画像に写っている物体に対応した，リストのインデックスのことである．アノテーションをした結果は，識別モデルの学習時に使用する．

一般的にアノテーションは，手作業で行うものである．しかし，数千枚の画像を手作業で行うことは難しいので，自動で行えるようにした．分類モデル用のデータセットでは一枚の画像に電車が一つだけ映っている．その画像を配布されている識別用モデルで識別することでクラス番号と電車が映っている座標情報をテキストに書き込む．その後，本プロジェクトで識別する車両タイプリストに対応するクラス番号を上書きすることで，分類モデル用のデータセットから識別モデル用のデータセットを作成した．

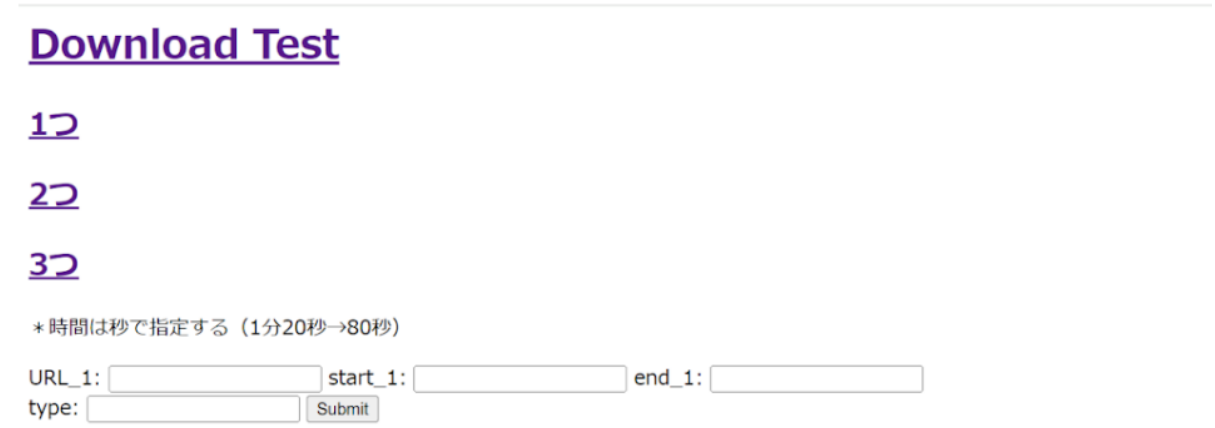


図 3.1: 動画のダウンロード 1

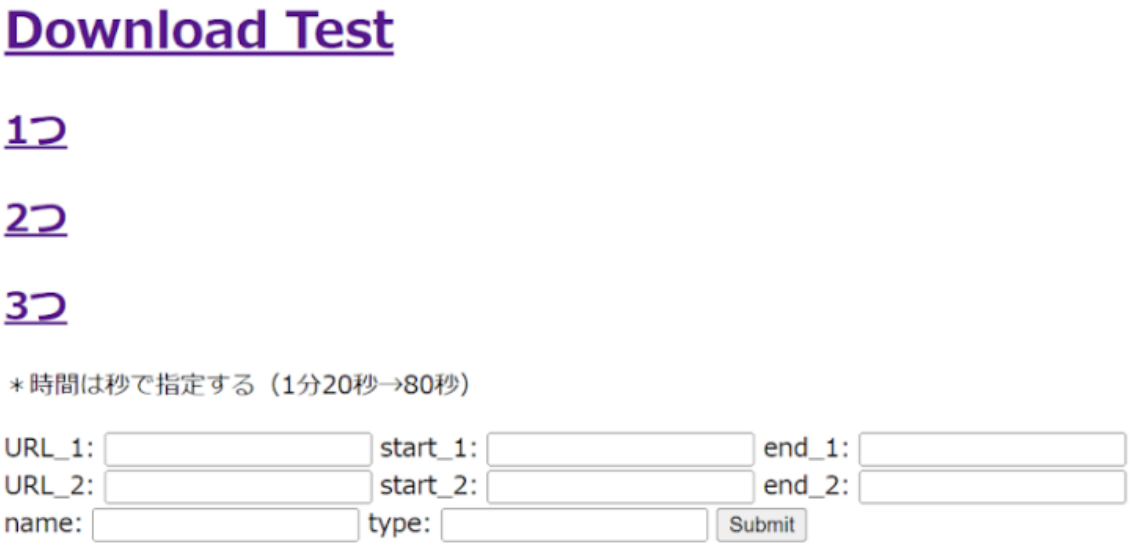
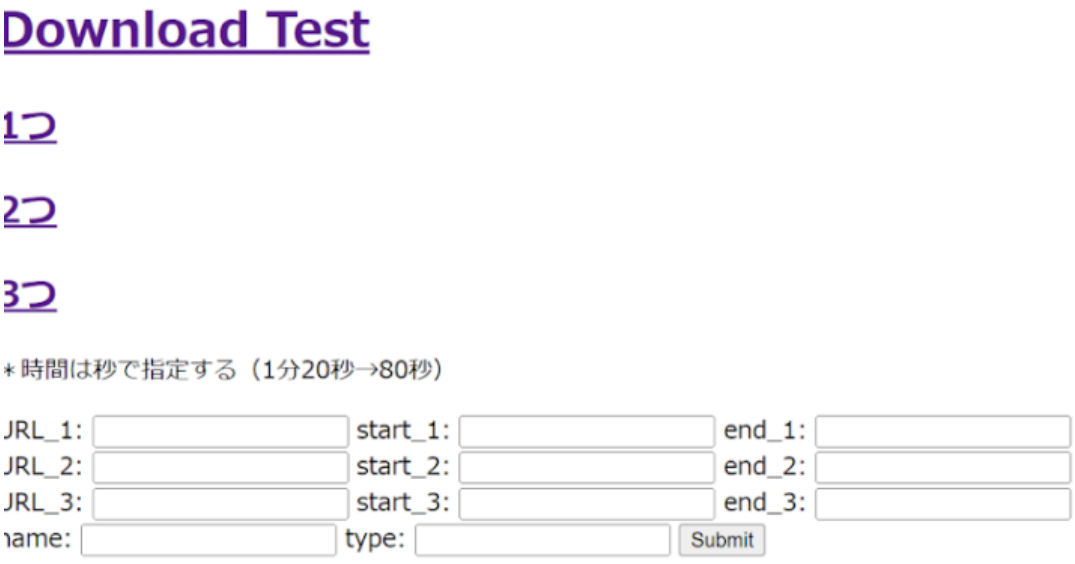


図 3.2: 動画のダウンロード 2



第 4 章

車両タイプ判別モデルについて

4.1　この章で書くこと

- yolo とは
- 学習の実行
- 性能評価
- 作成したモデルの使い方
- 出力されるものについて

4.2　YOLO とは

YOLO の説明
YOLO とは You Only Look Once の略で，人間のように一目見るだけで物体検出ができることを指している．データセットを作成し学習させることで，任意の物体のみ検出させることが可能である．
YOLOv8 の説明
YOLOv8 は YOLO シリーズの最新バージョンであり，ディープラーニングとコンピュータビジョンの最先端の進歩に基づいており，速度と精度の麺で比類のない性能を提供している．

4.3　学習の実行

学習時にモデルの性能がそれまで以上に向上しなくなると学習が強制終了するため，エポック数という学習用データを何回繰り返して学習させるのかを表す数を１０万回程度に設定して学習を進めた．強制終了した際のモデルが使用したデータセットで作れる最高の性能のモデルとなる．

4.4　作成したモデルの使い方

識別モデルと分類モデルで使い方はほとんど同じ作成したモデルをロードして，モデルに画像または動画を渡すと識別または分類をすることができる
識別

```
$from ultralytics import YOLO
model = YOLO("作成した識別モデルのパス")
results = model.predict("画像のパス", save = True)
$
```

分類

```
$from ultralytics import YOLO
model = YOLO("作成した分類モデルのパス")
results = model("画像のパス", save = True)
$
```

```
image 1/170 /home/nozaki/yt-dlp/test/detect_test/205_1.jpg: 384x640 1 205, 44.8ms
image 2/170 /home/nozaki/yt-dlp/test/detect_test/205_10.jpg: 448x640 1 205, 43.2ms
image 3/170 /home/nozaki/yt-dlp/test/detect_test/205_2.jpg: 448x640 1 205, 3.3ms
```

図 4.1: 出力されるもの

図 4.2: 分類モデルの評価

4.5　出力されるもの

１７種類の車両が各１０枚ずつ，合計１７０枚のテストデータセットを識別した際にターミナルに出力されるものを図 4.1 に示す．左端から，「何枚目か」「画像のパス」「入力画像のサイズ」「予想クラス番号」「予想クラス番号に対応する車両タイプ」「識別にかかった時間」が画像ごとに出力される．識別時に save_txt = True を追加すると画像ごとに予想クラスと座標情報がテキストファイルで出力される．

4.6　性能の評価

4.6.1　分類モデルの評価方法

１７種類の各車両の画像をそれぞれ１０枚ずつ集めて画像の分類を行った．分類モデルの予測した車両タイプと正解の車両タイプがどの程度あっているのか確かめることで性能の評価を行った．

4.6.2　分類モデルの評価

分類モデルの評価を図 4.2 に示す．縦軸が予測した車両タイプ，横軸が正解の車両タイプとするグラフである．

4.6.3 識別モデルの評価方法

作成したモデルは，IoU という指標で評価をする．作成したモデルが車両タイプ A だと認識した領域と車両タイプ A の領域がどの程度重なっているのか，を表す指標である．0.0 ≦ IoU ≦ 1.0 の範囲で表され，IoU は 1.0 に近いほど正確に予測ができていることがわかる．モデルが予測した領域と答えの領域が完全に一致しているときの IoU は 1.0 となり，全く重なりがない場合の IoU は 0.0 である．識別失敗数とは間違ったクラスだと判断した数であり，識別不能数とはどのクラスにも当てはまらないと判断した数とする．

4.6.4 識別モデルの評価

作成した識別モデルを使用し，テストデータセットの識別をした結果を表 4.1 に示す．
NaN とは識別成功数または識別失敗数が 0 のときに，IoU の平均が出せない場合の値である．

車両タイプ	識別成功数	IoU(平均)	識別失敗数	IoU(平均)	識別不能数
205	8	0.949791	1	0.913000	1
207	4	0.941828	2	0.940226	4
213	2	0.970088	7	0.923738	1
221	10	0.955405	0	NaN	0
223	8	0.970467	1	0.824151	1
225	1	0.688767	7	0.941395	2
227	4	0.953520	4	0.948728	2
271	5	0.844484	4	0.953993	1
281	2	0.838358	8	0.938570	0
283	7	0.953823	0	NaN	3
285	0	NaN	8	0.897906	2
287	5	0.960545	5	0.948412	0
321	8	0.932078	1	0.946316	1
323	9	0.954252	0	NaN	1
381	8	0.905294	1	0.950514	1
521	9	0.952872	1	0.810363	0
683	7	0.955607	3	0.962517	0

表 4.1: 識別結果の表

第 5 章

結論

あわれといふも，なかなか疎かなり．されば，人間の儚き事は，老少不定のさかいなれば，誰の人も早く後生の一大事を心にかけて，阿弥陀仏を深く頼み参らせて，念仏申すべきものなり．あなかしこ，あなかしこ．あわれといふも，なかなか疎かなり．されば，人間の儚き事は，老少不定のさかいなれば，誰の人も早く後生の一大事を心にかけて，阿弥陀仏を深く頼み参らせて，念仏申すべきものなり．あなかしこ，あなかしこ．あわれといふも，なかなか疎かなり．されば，人間の儚き事は，老少不定のさかいなれば，誰の人も早く後生の一大事を心にかけて，阿弥陀仏を深く頼み参らせて，念仏申すべきものなり．あなかしこ，あなかしこ．あわれといふも，なかなか疎かなり．されば，人間の儚き事は，老少不定のさかいなれば，誰の人も早く後生の一大事を心にかけて，阿弥陀仏を深く頼み参らせて，念仏申すべきものなり．あなかしこ，あなかしこ．あわれといふも，なかなか疎かなり．されば，人間の儚き事は，老少不定のさかいなれば，誰の人も早く後生の一大事を心にかけて，阿弥陀仏を深く頼み参らせて，念仏申すべきものなり．あなかしこ，あなかしこ．あわれといふも，なかなか疎かなり．されば，人間の儚き事は，老少不定のさかいなれば，誰の人も早く後生の一大事を心にかけて，阿弥陀仏を深く頼み参らせて，念仏申すべきものなり．あなかしこ，あなかしこ．

参考文献

[1] V.D. Vaughen and T.S. Wilkinson, ”System considerations for multispectral image compression designs,” IEEE Signal Process. Mag., vol.12, no.1, pp. 19-31, Jan. 1995.

[2] A. Said and W. Pearlman, ”An image multiresolution representation for lossless and lossy compression,” IEEE Trans. Image Process., vol.5, no.9, pp.1303-1310, Sept. 1996.

[3] 小野文孝, ” 静止画符号化の新国際標準方式（JPEG2000）の概要,” 映像情報メディア学会誌, vol.54, no.2, pp.164-171, Feb. 2000.

[4] D. Tretter and C.A. Bouman, ”Optimum transform for multispectral and multilayer image coding,” IEEE Trans. Image Process., vol.4, no.3, pp.296-308, March 1995.

[5] F. Amato, C. Galdi and G. Poggi, ”Embeded zerotree wavelet coding on multispectral images,” IEEE Proc. ICIP 97, vol.1, pp.612-615, 1997.

[6] B.R. Epstein, R. Hingorani, J.M. Shapiro and M. Czigler, ”Multispectral KLT-wavelet data compression for Landsat thematic mapper images,” Proc. Data Compression Conf., IEEE Computer Society Press, pp.200-205, 1992.

[7] J.M. Shapiro, S.A. Martucci, and M. Czigler, ”Comparison of multispectral Landsat imagery using the embeded zerotree wavelet(EZW) algorithm,” DLPO at Image Compress. Appli. & Innovation Workshop, pp.105-113, March 1994.

[8] J.A. Saghri, A.G. Tescher, and J.T. Reagan, “Practical transform coding of multspectral imagery,” IEEE Signal Process. Mag., vol.12, no.1, pp.32-43, Jan. 1995.

[9] J. Lee, “Optimized quadtree for Karhunen-Loeve transform in multispectral image coding,” IEEE Trans. Image Process., vol.8, no.4, pp.453-461, April 1999.

[10] B. Brower, B. Gandhi, D. Couwenhouven and C. Smith, “ADPCM for advanced LANDSAT downlink applications,” in Proc. 27th Asilomar Conf. Signals, Systems and Computers, Nov. 1993.

[11] N.D. Memom, K. Sayood and S.S. Magliveras. ,”Lossless compression of multispectral image data,” IEEE Trans. Geosci. and Remote Sensing., vol.32, no.2 , pp.282-289, March 1994.

[12] S. Gupta and A. Gersho, “Feature predictive vector quantization of multispectral images,” IEEE Trans. Geosci. and Remote Sensing., vol.30, no.3, pp.491-501, May 1992.

[13] K. Irie and R. Kishimoto, “A study on perfect reconstructive subband coding,” IEEE Trans. Circuits Syst. Video Technol., vol.1, no.1 , pp.42-48, March 1991.

[14] 小松 邦紀, 瀬崎 薫, 安田 靖彦, “濃淡画像の可逆的なサブバンド符号化法”, 信学論（D-II）, vol.J78-D-II, no.3, pp.429-436, March 1995.

[15] A.R. Calderbank, I. Daubechies, W. Sweldens and B. Yeo, ”Lossless image compression using integer to integer wavelet transforms,” IEEE Proc. ICIP 97, vol.1, pp.596-599, 1997.

[16] F.A.M.L. Bruekers and A.W.M.V.D. Enden, “New networks for perfect inversion and perfect reconstruction,” IEEE J. Sel. Areas Commun., vol.10, no.1, pp.130-137, Jan. 1992.

[17] 小松 邦紀, 瀬崎 薫, “可逆的離散コサイン変換とその画像情報圧縮への応用,” 信学技報, vol.IE97-83, pp.1-6, Nov. 1997.

[18] K. Komatsu and K. Sezaki, ”Design for lossless block transforms and filter banks for image coding,” IEICE Trans. Fundamentals, vol.E82-A, no.8, pp.1656-1664, Aug. 2000.

[19] 福間 慎治, 岩橋 雅宏, 神林 紀嘉, “可逆的色変換を用いた色彩画像の可逆符号化,” 信学論（D-II）,

vol.J81-D-II, no.11, pp.2680-2684, Nov. 1998.

[20] T. Nakachi, T. Fujii, and J. Suzuki,"A unified coding algorithm of lossless and near-lossless color image compression," IEICE Trans. Fundamentals, vol.E83-A, no.2, pp. 301-310, Feb. 2000.

[21] 仲地 孝之, 藤井 竜也, "可逆 KL 変換を用いた病理顕微鏡画像符号化法の研究," 2000 信学総大, D16-13, March 2000.

[22] 鷹合 大輔, 武部 幹, "可逆 WT・KLT を用いるマルチスペクトル画像の情報圧縮," 信学論（A）, vol.J84-A, no.3, pp.1-11, March 2001.

[23] 尾上 守夫, "画像処理ハンドブック," 昭晃堂, pp.554-561, 1987.

[24] 鷹合 大輔, 武部 幹, "マルチスペクトル画像のバンド間・バンド内相関除去による可逆情報圧縮," 第 15 回 DSP シンポジウム講演論文集, pp.475-48, Nov. 2000.

[25] 鷹合 大輔, 武部 幹, "ウェーブレット変換を用いたマルチスペクトル画像の情報圧縮符号化法の研究," 平成 10 年度金沢工大卒業論文, 1998.

[26] 酒井 幸市, "デジタル画像処理入門," コロナ社,1997.

[27] 榊原 進, "ウェーブレットビギナーズガイド," 電機大出版局, 1995.

[28] 武部 幹, "情報圧縮、通信と回路理論," 信学技報, IT97-40, July 1997.

[29] Gilbert Strang and Troung Nguyen, " Wavelet and filter banks,' 'Wellesly Cambridge Press, 1996.

[30] 貴家 仁志, "よくわかるディジタル画像処理," CQ 出版社, 1996.

[31] " 金沢の暮らし", http://www.kanazawa-it.ac.jp

[32] 山田 太郎, " 金沢の一人暮らし", トンチンカン出版, 2016.

付録 A

開発したプログラム

A.1　セットアップ方法

にプログラムの使い方，セットアップ方法などを書きましょう．ここにプログラここにプログラムの使い方，セットアップ方法などを書きましょう．ムの使い方，セットアップ方法などを書きましょう．ここにプログラムの使い方，セットアップ方法などを書きましょう．

```
$ sudo apt-get install python3-pip      # PIP コマンドの導入
$ echo "Hello WOrld"
Hello WOrld
```

にプログラムの使い方，セットアップ方法などを書きましょう．ここにプログラここにプログラムの使い方，セットアップ方法などを書きましょう．ムの使い方，セットアップ方法などを書きましょう．ここにプログラムの使い方，セットアップ方法などを書きましょう．

A.2　使い方

ここにプログラムの使い方，セットアップ方法などを書きましょう．ここにプログラムの使い方，セットアップ方法などを書きましょう．ここにプログラここにプログラムの使い方，セットアップ方法などを書きましょう．ムの使い方，セットアップ方法などを書きましょう．ここにプログラムの使い方，セットアップ方法などを書きましょう．ここにプログラムの使い方，セットアップ方法などを書きましょう．

A.3　ソースコード

リスト A.1: サンプルプログラム

```
1  #include<stdio.h>
2  int main(){
3    return 0; // ウギャー！
4  }
```

リスト A.2: スパゲッティソース

```
1  #! /usr/local/bin/ruby -Ks
2  # numbers.rb
3  print "正の整数値を表す文字列を入力してください。正の整数値を表す文字列を入力してください。\n"
4  while true
5    print ">"
6    line = gets.chomp # 改行コードを切り捨てる
7    break if line.empty?
8    begin
9      v = Integer(line) # 文字列を整数化する
10   rescue
11     puts "変換できません。"
12     next
13   end
14   printf ("2進法：%b\n",v)
```

```
15    printf ("8進法：%o\n",v)
16    printf ("10進法：%d\n",v)
17    printf ("16進法：%x\n",v)
18  end
19  puts "Bye."
```

付録 B

いいいいい

　あああああああああああああああああああああいいいいいいいいいいいいいいいいいいいいいいいいいいい
いううううううううううううううううう