A Patch-Based Neural Network Super-Resolution for Low-Delay Real-Time Processing

博 士 論 文

低遅延リアルタイム処理に適した
パッチ型ニューラルネットに基づく
超解像に関する研究

A Patch-Based Neural Network Super-Resolution for Low-Delay Real-Time Processing

金沢大学大学院 自然科学研究科
電子情報科学専攻

学　籍　番　号：1624042006
氏　　名：青木　玲央
主任指導教員名：今村　幸祐
提　出　年　月：2019 年 9 月

目　次

第 1 章 序論

1.1 本研究の背景

1.1.1 高解像度化は時代の潮流

我が国における地上デジタル放送は 2003 年から試験放送が始まり，2006 年の暮れには全ての都道府県庁所在地での受信が可能となった [1-1]．ちょうどその頃から，かつては一般的であった VGA 程度の解像度を表示できるブラウン管テレビから，Full-HD 相当の解像度を表示できる液晶，もしくはプラズマといった新方式の薄型テレビへの置き換えが行われていった．総務省の統計では 2007 年には 19.3%だった薄型テレビの世帯普及率は 2010 年には 75.3% まで増加したという記録もある [1-2]．また，映像データを記録する媒体も，かつては VTR（ Video Tape Recorder ）と呼ばれた磁気テープ式のものが一般的であったが，420p を記録できる DVD（ Digital Versatile Disc ）と呼ばれる光ディスク式が 2003 年ごろから本格的に利用され[1-3], 2006 年頃から今日にかけては 1080p まで記録できる BD（ Blu-ray Disc ）が広く普及するまでに至った．さらに，近年若年層を中心に幅広く普及したスマートフォンで有名な iPhone では，2010 年製のモデル（ iPhone 4 ）から画素の細かさが人間の目で識別できないレベルにまで到達したことを宣言し，Retina Display という言葉が世間を騒がせた．その後，iPhone 6 以降では 1080p を凌ぐ解像度が手に収まる小型ディスプレイで実現できるまでに至っている．

これら技術革新を通して，映像信号の高解像度化，或いは，高精細化は映像の高画質化を果たす上での重要なファクターとして人々に広く認識され，最近ではさらに高解像度である 4K やスーパーハイビジョンとも称される 8K も現実になってきた．例えば総務省では，ICT 政策の一環として 4K/8K の推進をしており，2020 年の東京オリンピックでの中継を 4K/8K 放送で実現すべく，既に試験放送を開始している [1-4]．また，高解像映像コンテンツの民間提供に関しては，放送業界のみにとどまらず YouTube [1-5] や Netflix [1-6] 等で知られるインターネット上の動画配信サイトでも多く見られ，4K 映像に関しては既にサービスが始まっているところも多い．このように 2000 年代前半から始まったこれら映像の高解像度化は映像の高画質化に向けた時代の潮流であり，映像撮影，編集，表示，配信サービスに至る様々な産業分野において，近年取り組まれ続けている重要なテーマの１つである．

また，映像信号の高解像度化とはすなわち，映像に含まれる情報量の増加を意味しており，単に視覚的な審美性に留まらない実用的な動機に基づいて検討されているケースが最近では増えてきている．例えば，内視鏡手術の現場においては，より高精細な映像を得ることで人の視力を越えた細部まで観察することが可能となり，患者の負担が少ない精密医療（ Precision Medicine ）に近づくことが出来ると言われている．また，監視・セキュリティの分野では，より高解像度な映像を得ることで，画像中の注目したい部分

の拡大や，カメラ設置数を削減させるといった効率化やコスト削減も実現できる．さらには，近年，急速に発展した AI（Artificial Intelligence）技術を用いれば，高精細な画像から不審者を検出し，犯罪を未然に防ぐことも出来る可能性も出てきている．これらは当然ながら，映像が高精細であるほど検出精度を高めることが出来る．このように，映像の高解像度化に対する需要は多種多様に存在し，かつ，今後の技術革新によって，その重要性はさらに高まっていくことが予想される．

1.1.2 リアルタイム超解像の需要

映像の高解像度化が社会的に望まれる一方で，撮像デバイスや伝送系の帯域等の制約上，高解像度映像システムの実現が容易でないケースもある．例えば，胃カメラ等でよく使われる経鼻内視鏡では，患者への負担を軽減するため，カメラヘッドはなるべく小さくしたいという要望がある．それはすなわち，カメラ素子の構成要素である光学レンズモジュールや CMOS/CCD 等のイメージセンサの大きさに物理的な制約があることを意味し，高解像度化への大きな阻害要因となっている．なぜなら，イメージセンサの大きさを一定にしたまま高解像度化を実現する場合，1 画素あたりの受光面積が解像度の増加に伴って小さくなるため，暗電流の影響により S/N 比が低下し，結果的に画質劣化が生じやすくなるからである．従って，画質劣化をさせずに高解像度化を実現するには一般的には感度の高いイメージセンサを開発するか，より明るいレンズを検討するか，もしくはより明るい照明機材を検討するかといったアプローチが検討されるが，いずれのアプローチもやはり人体への熱的・スケール的な影響という制約の中でトレードオフが発生してしまうのが現状である．それ故，撮像系の性能にもやはり限界が存在し，「超解像」のような映像の高解像度化，或いは，高画質化に関する機能が必要とされている．

また，内視鏡を用いた手術の場合では明らかなように，映像処理にリアルタイム性が重要視される場合がある．なぜなら，開腹しながら行う外科手術とは異なり，医師は内視鏡映像を見ながら患部を切除したり，糸を縫い合わせたりといった施術をするからである．つまり，表示映像は医師の手元の動きに遅延無く追従しなければならず，そうでなくては執刀医への負担だけでなく，正確な医療への妨げとも成り得る．このようなリアルタイム超解像に対する需要は内視鏡手術の例だけでなく，様々なシーンで存在する．例えば，自動車の自動運転を実現していく上での判断は瞬時に行う必要があるが，周辺の 3 次元情報である奥行き情報（ Depth Map ）のセンサー解像度は一般に画像である RGB 信号に比べて低く，高解像化による情報補間をする場合は，やはりリアルタイムで行う必要がある．また，映像制作等の現場においては，高解像度コンテンツの生成やその保持には多大なコストが必要となるが，表示機器側でアップコンバートできれば，コスト削減につなげることができる．ただし，そのためにはリアルタイムに伝送される映像信号に追従して変換しながら表示する技術も同時に必要となる．

以上のように，物理的な制約によって高解像度化が困難なケースや，既に出来上がった映像コンテンツを視聴したいケースなどでは，表示機器側，或いは画像認識の前処理として，リアルタイムに超解像を実現することに意義がある．さらに，その場で得られ

た信号をリアルタイムで処理できるため，処理後のデータを保持する必要もなく，ストレージやメモリを削減するといった効果も期待できる．しかしながら，先に述べたような 4K/8K といった高解像度画像をリアルタイムで処理するには，解像度に応じた大規模なパイプライン型の並列演算回路が一般的には必要であり，映像処理のアルゴリズムによっては遅延や回路規模の観点で実現不可能となる場合がしばしばある．これが本研究の目的でもあるコスト削減，低遅延化へのモチベーションに繋がっている．

1.1.3 表示機器における「超解像」

まず，表示機器における「超解像」の定義はあいまいである．単に解像度を上げるという概念からすれば，拡大補間処理もまた超解像と呼べるかもしれないが，一般的に表示機器で拡大処理として実装されることが多い線形補間や Bicubic 補間を「超解像」と呼ぶことは少ない．また一方で，入出力の解像度に変化が無くても「超解像」と呼ばれることもあり，寧ろ，そのケースの方が多い．これは入力画像が対象解像度に対して十分な情報を有していない「低品質な画像」であるのに対し，出力画像がより「高品質な画像」に近づいているという意味での超解像である．さらに言えば，超解像にも複数枚の低解像度画像を組み合わせて 1 枚の高解像度画像を生成する，所謂，「複数枚超解像」と呼ばれるものと，1 枚の低解像度画像（或いは，低解像度品質の画像）から何らかの仮定に基づいて推定された高解像度画像（或いは，高解像度品質の画像）を生成する「1 枚超解像」があり，超解像と一口に言ってもその定義にはさまざまなものがある．

実際，表示機器メーカーの製品リリース情報をみても「超解像」という言葉には揺らぎがある．例えば，東芝は 2008 年に発売した液晶テレビ REGZA ZH7000 で世界初の超解像技術搭載［1-7］と謳っていたが，当時は超解像といえば日立製作所が開発していた同一の対象物に対して僅かに視点が異なる画像を複数枚用意することで，サブピクセルレベルでの位置あわせを行い，それらを合成した結果から解像度の高い画像を生成する複数枚超解像の印象が強く，この超解像はどういう意味なのかという物議を醸した．そのため超解像とは一体何なのか，という議論もあった．その影響もあってか，単に差別化を狙ってか，IO データが 2011 年に発売した液晶ディスプレイ（ LCD-MF274XBR ）では，「美解像」という名称がつけられていた［1-8］．また，2011 年に EIZO から発売された FS2332 では「Smart Resolution」と呼ばれる名称で 1 枚超解像に関する技術がリリースされた［1-9］．いずれの技術も現在の感覚からすれば超解像に分類されるものと推察されるが，各メーカーが出す超解像の手法はそれぞれ独自のものであり，またその細部はブラックボックスとなっていた．また，学術的な論文とは異なり，映像復元の正確性よりも見た目の印象を重視した画作りとしての性質が多分に含まれていた．

一方で，表示機器と一口に言っても，主に鑑賞目的として映像を表示するテレビと内視鏡手術等の特殊用途向けのディスプレイでは対象としているユーザーとその使用目的の違いから，映像処理技術の開発においても技術的な思想や考慮すべき要件が異なる．例えば「遅延」という要素はテレビ放送ではそれほど重要視されていない．仮に表示映像が複数フレーム遅延したとしても音声さえ同期していれば問題はなく，製品によっては映像の遅

延に合わせて音声も遅延させている．しかしながら，前節で例に出した内視鏡手術の場合には，映像と手元の感覚が一致しなければ手技に影響が出るため，一般に表示機器内でのフレーム遅延は望ましくない．また，色味や質感についても，テレビの場合は若干鮮やかな色彩や質感のほうが精彩を放つため好まれる傾向にあるが，表示された内臓の色や質感で患部の状態を見極める内視鏡手術の現場では，そのような脚色はむしろ不要で，まるで開腹手術をしていた時のような，より正確な色再現，もしくは，各内視鏡メーカーが拘る色表現が忠実に再現されることが望まれる．例えば，内視鏡メーカーによっては黄色を強調して脂肪を認識しやすくするといった画作りがあり，実物とは異なるが，術者の効率化を図るものである．また，こういった色表現は各内視鏡メーカー固有の特徴を表すことも多いため，表示機器としてはあくまで入力された映像に対して忠実な映像再現が求められるケースが多い．このような事情もあり，特定用途向けのディスプレイでは極力遅延の少ない超解像処理が求められ，かつ，画作りというよりも入力信号により忠実な色の再現性が求められる傾向にある．また，この思想は超解像を評価する上においても重要な視点であり，遅延が少なく，かつ，復元性能の高い自然な映像再現であるという点が重要になってくる．

1.1.4 「超解像」の技術動向と課題

学術的な観点で言えば，どの技術が超解像に該当するかは表示機器に搭載された機能と比べれば比較的分かりやすい．例えば，静止画処理，動画処理等を包括するトップカンファレンスとして知られる ICIP（International Conference on Image Processing）等の国際学会では「Super Resolution」という分かりやすいセッションが存在し，論文中にも Super Resolution とはっきり表記されている場合が殆どである．そして 2000 年代後半から始まった各メーカーの超解像技術もやはり学術分野のトレンドに影響されている部分があり，端的に言えば，超解像が出始めた当初は解像度の異なる画像で視覚特性的に最も違いを実感しやすいエッジ成分の復元に注力したものが多かったが，2010 年代以降では，質感（テクスチャ）再現に重点を置いた技術がトレンドとなっていった．そして，学術分野において近年注目が高まっているのが，Deep Learning を活用した超解像手法であり，その先駆けとなったのが 2014 年の ECCV（European Conference on Computer Vision）で発表された SRCNN（Super Resolution Convolutional Neural Network）[1-10] である．

関連研究に関する詳細は 2 章で述べるが，ここでまず強調したいのは，従来，技術者が経験的に設計してきた手法と比較すると，非常に単純な全く異なるアプローチである End-to-End 学習という手法で機械学習されたニューラルネットワークが従来手法を上回る性能に到達した点である．つまり，SRCNN では 3 層の畳み込みニューラルネット（Convolutional Neural Network: CNN）で構成されるネットワークの入力に低品質画像を入力し，その出力が教師信号である高品質画像に近づくように誤差逆伝播法（Back Propagation）によって内部パラメータを修正するという行為を繰り返し行うだけで，内部パラメータが自然と最適化され，未知の低品質画像を入力した際も高い精度で高品質画像を生成出来るのである．この考え方は内部を完全にブラックボックスとして扱い，末端

の入力と出力のみでニューラルネットが形成されることから End-to-End と呼ばれるが，従来の様々な仮定条件によって複雑化していった超解像技術設計の考え方をまるでリセットするかのような衝撃を与えた．

また，もう１つ着眼すべき点は，３層ニューラルネットに含まれる演算はすべてフィードフォワードで構成されているという点である．一般に超解像やノイズ除去といったいわば非可観測な真値を推定する不良設定問題では一意解が存在しないため，単純な計算による解析解の導出は出来ない．そのため，様々な仮定を前提とした高度な数式を定義し，反復演算を前提とした最適化手法による近似解を導くことが高性能な超解像手法では一般的なアプローチとなっていた [1-11]．しかしながら，SRCNN は非常に単純なフィルター処理と非線形演算である活性化関数をたかだか 3 層組み合わせただけで高度な推定を実現でき, かつ, そこには反復等の演算が必要とならない. つまり, 理論的には ASIC（Application Specific Integrated Circuit）や FPGA（Field-Programmable Gate Array）などの専用ハードウェアで必要な計算素子を全て並べることが出来れば，少なくともリアルタイム処理が可能となることを意味している．従って，処理がフィードフォワードであるという点は，内視鏡や安全監視等といった特定用途向けディスプレイへの応用を考える上では非常に優れた性質といえる．

しかしながら，SRCNN を実際に特定用途向けディスプレイの超解像として採用するには幾つかの課題が存在する．詳細は 2 章で述べるが，最も大きな課題はディスプレイの高解像度化に起因する膨大な計算量である．例えば，SRCNN を 4K 60fps で実現するには約 28T FLOPS もの計算量が必要となる．これは SRCNN が 3 層の畳み込み演算フィルターで構成されることから単純に導かれる．また，SRCNN の中でもコストパフォーマンスの高い SRCNN 9-1-5 の場合では第 2 層の演算量が激減するため, 約 4 T FLOPS の計算量が必要となる．これらの数値は現時点で非常に膨大な計算量であると言わざるを得ない．

現実的には 4K などの高解像度表示機器に対して GPU を実装することは，価格面，電力面，処理遅延，等の課題が生じるため，先にも述べた ASIC や FPGA といった専用ハードウェアでの実装が一般的なアプローチとなる．また，その場合のハードウェア実装コストは，上記フィルター演算に必要な演算回数ではなく，１画素を出力する上で必要となる必要最小限のハードウェア構成を考えるべきであり，結論だけを述べると，SRCNN 9-5-5 の場合は約 170 万回，SRCNN 9-1-5 の場合は約 18 万回の積和演算が必要となる．これらの計算量はやはり一般的な感覚からすると非常に大きい．例として，比較的大規模で高価格な FPGA として知られる Xilinx 製の Kintex UltraScale の中でも最高性能である KU115 を引き合いに出すと, 高精度な演算が出来る DSP スライスは 5,520 slice しか用意されていない．勿論，DSP で足りない部分は System Logics Cells で補うことになるが，それでも 145 万回路の Logic Cell での SRCNN 9-5-5 実装はまず不可能であり，やはりコスト面において課題があるという結論に達する．

1.2　本研究の目的

本研究では，Deep Learning が可能にした End-to-End 学習という優れた特性を保持しながらも実用的かつ高性能な超解像システムの構築を目指している．具体的には内視鏡用ディスプレイや駅のホーム監視用ディスプレイといった特殊な用途にも対応できるようにするため，低遅延性を兼ね備えながらも SRCNN と同等，あるいはそれ以上の復元性能を持った超解像システムの構築が本研究の主たる目的である．

計算コストの観点でみると，SRCNN の登場以降，学術分野における超解像研究の主流は Deep Learning による事前学習・推定型超解像に移っており，それらの多くは CNN をベースにした多層構造を前提としているため計算量が増大していく傾向にある．例えば，VDSR [1-12] はその代表ともいえる手法であるが，20 層もの CNN を重ねることで更なる性能向上を果たした一方で，その計算量は SRCNN 9-5-5 の約 200 倍にまで膨れ上がっている．これは CNN を重ねることによってより幅広く領域を見渡して処理を行っていることに相当する．広範囲を見渡すことによって情報が増え，精度の向上が期待できるが，その反面，拡がってしまった受容野の多くのパターンを処理する必要性がでてくるため，さらなる計算コストが要求されており，そのままリアルタイム処理への適用することは現実的ではない．本研究では，低遅延でリアルタイム処理が可能なニューラルネットによる超解像システムを目指す．

品質の観点では，詳細は 2 章で述べるが，これまでの代表的な超解像である SRCNN では予めバイキュービック補間で拡大された入力画像に対し，拡大率の周期で生じる補間画素位置を考慮せずに同一の CNN によって推定を行っていたため，入力画像に含まれるジャギーやリンギングといったアーチファクトを抑制しきれないといった課題も存在する．本課題に対し，提案手法ではパッチベースの全結合型ニューラルネットを採用することで，補間画素位置に対応したパッチ抽出を行い，小規模なニューラルネットワークでも主観的な画像品質の向上が実現できることを目指す．

また，拡張性の観点から，表示機器としてより幅広いシチュエーションに対応できることを考慮して，SRCNN と同様に解像度変換を伴わない純粋な画質改善効果のみを備えた超解像機能の実現を目指す．これにより，入力機器側で既に簡易な補間によって解像度を高めてから表示機器に映像が入力される簡易なアップスケール機能と超解像機能を分離することが可能になる．つまり，単純な拡大処理行うだけならコストの安い汎用的な前段スケーラに任せることができるため，付加価値を上乗せする部分は後段の FPGA，もしくは ASIC で賄うといった戦略が取れることになる．結果として，拡大機能の実現に必要となる DRAM（Dynamic Radom Access Memory）を超解像機能の有効・無効に関わらず，前段のスケーラに集約することができ，コスト削減につなげることができる．これは単に DRAM の部品コスト削減というだけでなく，基盤面積，必要 I/O 端子，消費電力，発熱，輻射等，実際に商品設計・開発を進めていく上での様々なメリットに繋がる．

上記の目標達成のため，本研究では低遅延でリアルタイム処理が可能なパッチベースの全結合型ニューラルネット構造による超解像システムを提案する．パッチベースとは画像を小領域のパッチ画像に分割して処理する手法であり，入出力の画像サイズがパッチサイ

ズに限定されるため，CNN とは異なり受容野を拡げずにディープな推定が可能となる．また品質向上のために補間画素位置に基づく学習対象の限定化を行う．これにより補間画素位置とニューラルネットの各入力ノードの対応関係が定まるため，限られたリソースでも処理画像が主観的に向上する．システム全体としては，超解像の中核を担う入力パッチ画像から対象パッチ画像の高周波成分を推定する全結合型ニューラルネットと，リアルタイム映像信号のラスタースキャンに合わせて対象パッチの抽出，及び，推定結果の合成を行う周辺部によって最終的な目的である，End-to-End 学習という優れた特性を保ちながらも軽量で高性能な超解像システムを実現する．

1.3　本論文の構成

本論文は，表示機器に適した低遅延でかつリアルタイム処理が可能なニューラルネット型超解像システムに関する研究成果を報告するものであり，以下の各章で構成される．

第 1 章 序論 では，映像の高解像度化への社会的期待が高まった背景と，リアルタイム処理の必要性を述べた．また，表示機器としての「超解像」とその技術動向の概略を述べ，その上で近年主流となってきた Deep Learning を用いた高性能な超解像システムを構築する上での課題について概要を述べた．さらに，本研究での目指す超解像システムの目標と概要を示した．

第 2 章 低遅延リアルタイム処理の課題と関連研究 では，まず，表示機器におけるリアルタイム映像表示システムの概要を説明し，本研究で目指す超解像システムの枠組みとそれを実現する上での課題について説明する．その上で，リアルタイム映像システムの構築という観点から従来の超解像技術を俯瞰し，当該研究領域における従来手法の課題を示す．

第 3 章 パッチベース型 DNN による超解像システム では，本研究の目的である低遅延でかつリアルタイム処理に適したニューラルネット型超解像システムを提案し，その手法について説明する．また，アーキテクチャの設計思想，学習による内部パラメータの最適化手段，さらには，任意サイズの画像を前提にした復元手法といった内容を記載する．

第 4 章 シミュレーションによる性能評価と解析 では 3 章で説明した手法に対する性能評価のための実験結果とその考察を示す．まず，復元性能については客観評価指標と視覚的な違いを主観的な視点で記す．さらに，各手法のハードウェア設計時におけるコストを試算し，コストパフォーマンスの比較を実施する．さらに，検討手法の解析として，補間画素の一致性に関する性能評価や，学習用データセットの変動による性能変化，さらには学習後に可能なさらなるコスト削減手法について紹介し，提案手法の有効性と今後の展望について述べる．

第 5 章：結論 では，各章の主旨を統括し，低遅延かつリアルタイム処理に適したニューラルネット型超解像システムの研究に関する本論文の結論を述べる．

参考文献

[1-1] 総務省，地上デジタル放送への完全移行について（平成 23 年 7 月）資料 26-4, p.2.

[1-2] 総務省，地上デジタルテレビ放送に関する浸透度調査の結果，平成 22 年 5 月 27 日.

[1-3] 橋本　慶隆, “映像記録メディアの変遷 -アナログからディジタルへ-“，尚美学園大学芸術情報学部紀要, Vol.5, pp.29-44, 2004.

[1-4] 総務省，放送政策の推進，平成 30 年度版情報通信白書，第 2 部，pp.314-323，2018.

[1-5] YouTube, https://www.youtube.com/

[1-6] Netflix, https://www.netflix.com/jp/

[1-7] 東芝 REGZA ZH7000, http://www.toshiba.co.jp/

[1-8] IO データ, LCD-MF274XBR, https://www.iodata.jp/product/lcd/wide/lcd-mf274xbr/

[1-9] EIZO, FS2332, https://www.eizo.co.jp/support/db/products/model/FS2332

[1-10] C. Dong, C.C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” European Conference on Computer Vision, pp.184–199, Springer, 2014.

[1-11] Shunsuke Ono, Isao Yamada; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4090-4097, 2014.

[1-12] J. Kim, J. Kwon Lee, and K. Mu Lee, “Accurate image super-resolution using very deep convolutional networks,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1646–1654, 2016.

第 2 章　低遅延リアルタイム処理の課題と関連研究

2.1　緒言

「超解像」と一口に言っても，実際には様々な方式があり，目的に応じてシステムとしての要件や枠組みも異なることは既に述べた．本章では，まず始めに本研究の目的である特殊用途向けディスプレイへの搭載という観点から，映像表示システムにおけるリアルタイム処理の枠組みを説明し，本研究で目指す「超解像」システムが備えるべき特性について，より具体的に説明する．その上で，映像表示システムの構築という観点から従来手法を俯瞰し，従来手法における課題を示していく．

2.2　映像表示システムにおける低遅延リアルタイム処理

2.2.1　ディスプレイにおける「超解像」の位置づけ

映像表示システムは，図 2.2.1 に示す Imaging Chain System において，カメラ等の撮像系から，伝送系を通った後に入力される表示系に位置する．通常，表示系に入力される方法としては, 民生ビデオ機器に多い HDMI（High-Definition Multimedia Interface）ケーブルやパソコン等に多い DP（Display Port）ケーブル，或いは，特殊用途でノイズ耐性も強い SDI（Serial Digital Interface）ケーブルなどが使用される場合が多い．これらの伝送ケーブルは各システムにおいて，様々な理由によって選定されている．例えば，手術室の場合は電気メスによる電磁パルスがノイズ発生源となるため，ノイズに強い SDI ケーブルが望まれる．また，SDI ケーブルで一般的な BNC 端子にはロック機能があるが，これは術中でも簡単に抜けないようにするための仕組みとして需要がある．一方，民生機器での使用が多い HDMI ケーブルでは, 足を引っ掛けた場合にケーブル側が簡単に抜ける仕組みになっており，映像の表示よりも人の転倒危険性軽減のほうが重要視されている．このように，目的や用途に応じて映像伝送の仕組みは様々であり，かつ，それぞれの伝送ケーブルには異なるビデオ規格が定められていることが一般的となる．

図 2.2.1 Imaging Chain System

上述の通り，用途・目的に応じて様々な伝送形態・通信規格が存在する伝送系に対し，ディスプレイとしては最終的に 1 つの映像を液晶パネル等に表示しなければならない．これが映像表示システムに与えられた最も大きな役割といえるが，一般的には液晶パネル等の表示デバイス（Display Device）は，特定の仕様に従った映像信号を入力しなければ表示できないという制約があり，入力された信号をそのまま接続すればよいというわけではない．そのため，一般的な映像表示システムとしては，図 2.2.2 に示したように入力部，フォーマット変換部，画像処理部，出力部を備えた構成となっている．

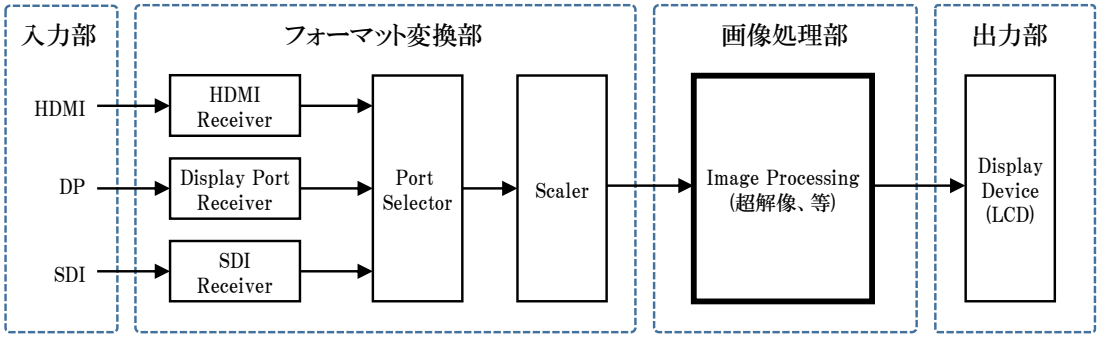


図 2.2.2 映像表示システムの内部構成

ここで，入力部は用途に応じた入力端子で構成される．具体的には，内視鏡用であれば SDI（ Serial Digital Interface ）, PC 用であれば DP（Display Port）といった具合で，需要に応じて入力ソースの種類や数は変動する．その後，各入力端子の映像信号はフォーマット変換部にある各伝送系規格に合った専用レシーバーに送られ，TMDS（Transition Minimized Differential Signaling ）等の汎用性の高いフォーマットに変換されて伝送される．次の Port Selector では，画面に表示すべき映像が選択される．通常は 1 つの入力ソースが選択されるが，場合によっては P in P（Picture in Picture）や P by P（Picture by Picture）など, 複数の映像信号を組み合わされた上で伝送される場合もある. そして，Scaler では入力された映像信号を液晶パネル等の表示デバイスに入力可能な形式に変換して出力する役割を担う．具体的には映像信号のタイミング調整や，解像度の変換，階調補正等が挙げられる．なお，フォーマット変換部の実装に関しては，特殊な要件がない限りは市販の汎用スケーラや，レシーバーを組み合わせることで実現されることが多い．そのため，特別な画像処理が必要とならない場合には，画像処理部をショートカットし，直接出力部の表示デバイスに入力されることもある．結局，超解像等の画像処理を挿入する位置としては Scaler と Display Device の中間が最も一般的で都合の良い配置となり，以下にその利点を列挙する．

① 画像処理部で対応すべき形式を限定できる

予めフォーマット変換部にてフォーマットが統一化されているため，フォーマットに応じた特殊な変換を考慮する必要が無くなることを意味する．具体的には，画像サイズやフレームレート，YUV444，YUV420 といった各種の映像信号フォーマットを個別に配慮する必要がなくなる．これは内部処理フローの単純化だけでなく，実装コストの削減にも繋がる．特に，画像サイズが予め統一化されているメリットは大きく，拡大処理に必要な回路や遅延時間を画像処理側のチップで新たに発生させる必要がなくなる．また，超解像機能を実現する視点から見ると，画像拡大処理と画像処理部を切り離すことを意味するので，予めバイキュービック等の手法で拡大された画像にも適用させることが可能となる．

② 画像処理部のカスタマイズが容易となる

求められる画像処理の種類は用途によって異なる．例えば，本研究のテーマである超解像が求められる場合もあれば，夜間の監視映像では明るさ補正やノイズ除去が求められる場合もある．このようなケースにおいても排他性の高い上記システム構成であれば，画像処理部のロジックを書き換えるだけで様々なニーズに対応できる．また，場合によっては特別な処理を必要としないケースも当然あり，その場合は画像処理部を取り除くことによって，システムを実現することが出来る．

以上のように，ディスプレイにおける「超解像」の実装箇所としては，拡大等の事前処理を施した後の画像処理部に配置するのがシステムの汎用性や効率性，実装コストの面から適している．以上の理由から，当研究では解像度を高めるのではなく，低解像度品質の画像から高解像品質の画像に補正する超解像手法を前提に検討を行った．

2.2.2 低遅延リアルタイム処理

「リアルタイム処理」という言葉の感覚は人によってイメージが異なる場合がある．例えば，カメラで撮影された映像を解析して，数秒間隔で人の位置を表示できる画像認識システムがあった場合，リアルタイムで処理をしていると呼べるかもしれない．しかしながら，いわば表示を司るディスプレイの画像処理という立場で「リアルタイム処理」という言葉を使う場合は，上記の例とは異なり，入力された画像信号を変換して出力する際に，そのスループットを維持していることを意味する．図 2.2.3 は Full HD として知られる 1080p 60Hz の画像信号を図式化したものであるが，一般的な画像伝送では，ブラウン管時代に規格化された NTSC（National Television System Committee）方式を模倣し，ラスタースキャンと呼ばれる方法で画素情報が伝送される．つまり，画面最上部左上の 1 画素から水平方向右側に向かって順に一画素ずつ情報を伝送し，1 ライン走査が終わると一定のブランク期間の後に次ラインの画素情報を送り始める，といった具合である．

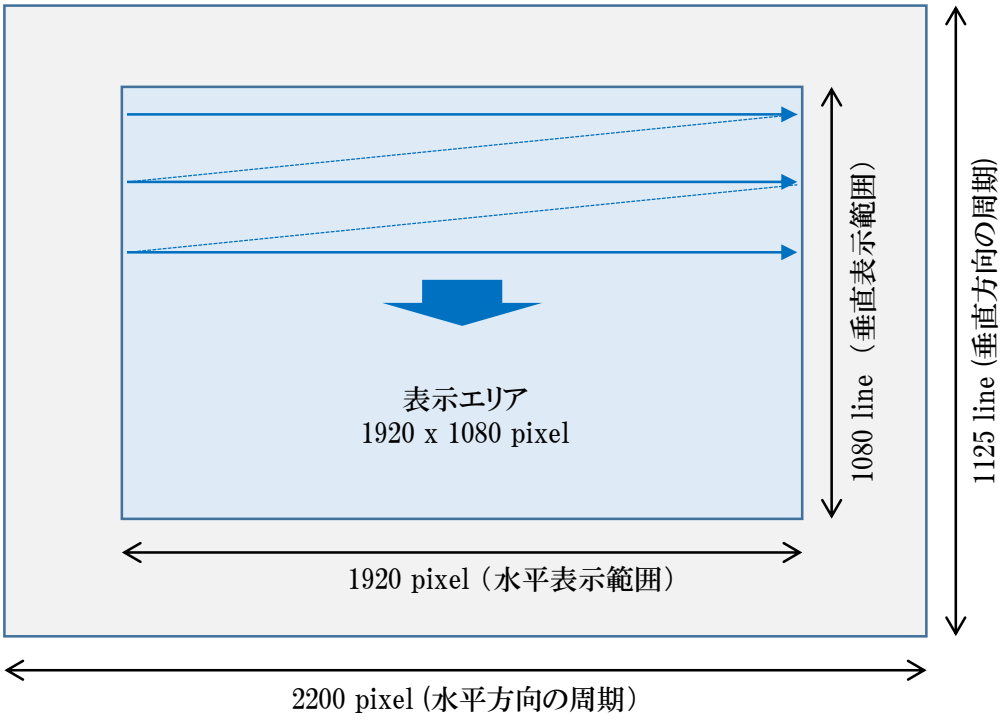
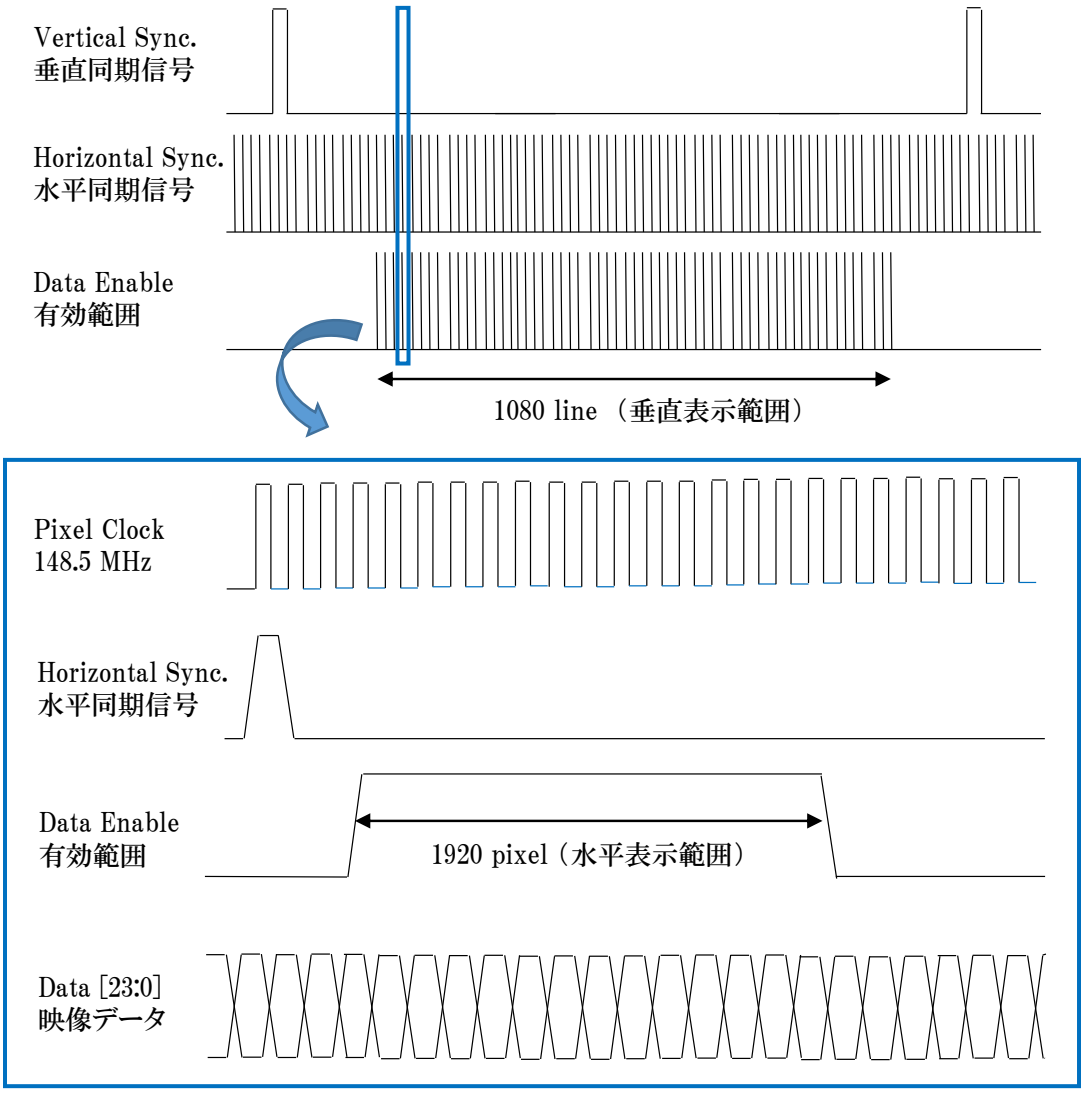


図 2.2.3: 画像信号の構成（Full HD）

図 2.2.4 は実際の信号波形をオシロスコープで見た場合をイメージして図に現したものであるが，垂直同期信号（Vertical Sync.）が立ち上がる度に新たなフレームの画像が送信され，水平同期信号（Horizontal Sync.）が立ち上がる度に新たなライン情報が送信される．また，図 2.2.4 の下側は 1 ラインの映像伝送期間を拡大したイメージであるが，148.5 MHz の Pixel Clock 信号（ 周期は僅か 6.7 nsec 程度 ）に同期して映像データが送られ，かつ，そのデータの有効範囲エリアを Data Enable 信号が示すといった構成になっている．つまり，映像処理におけるリアルタイム処理では，上記フォーマットで伝送されてくる信号を受けて計算を行い，同様なフォーマットで出力をする演算回路を持つ必要がある．



2.2.4: 画像信号の内部構成（Full-HD）

ここで，フレーム遅延のない「低遅延」リアルタイム処理とは，図 2.2.5 に示すようにその遅延システムが 1 フレーム未満であり，ラインオーダーの遅延時間で映像処理が完結できる処理のことを指す．つまり，低遅延リアルタイム処理を実現する超解像手法としては，以下の要件を満たさなければならない．

① 出力画素値の計算が一定時間（ラインオーダーの遅延）で完了できること．
② 出力画素値の計算に必要な情報がラインオーダーの範囲で収まっていること．

まず，①は一定のペースで画像信号を生成し続けるために必要な条件である．例えば先に例を挙げた顔認証の場合は，人数が多い場合は一時的に検出時間が長くなってもさほど問題は無いかもしれないが，映像表示システムの場合は常に一定のペースで画像信号を液晶パネル等の表示デバイスに送り続けなければならず，それが乱れることは画乱れを起こすことに直結する．これは反復演算のような最適化問題が含まれた映像処理をする場合には注意が必要で予め決められた試行回数で演算を打ち切るなどの工夫が求められる．

次に，②は図より自明であるが，出力する段階で入力されていない情報は使えない，という意味である．しかしながら，多くの場合の画像処理に関する論文手法は PC 上の処理を前提としており，この条件を満たしていない場合が多い．例えば，前章でも例を挙げた SRCNN［2-1］の標準的な処理手順では，画面全体にフィルターをかけた結果を中間結果として保持し，そこにまたフィルターをかけるといった処理が行われているが，画面全体に処理をかける時点で 1 フレーム分の画像を取得しなければならず，それ以上の遅延が発生してしまうことになる．従って，低遅延リアルタイム処理を実現するには，フレーム単位ではなく画素単位で処理方法を行う必要がある．つまり，対象出力画素を生成するために必要な周辺画素情報の範囲が処理遅延の下限を示す目安となる．なお，SRCNN の低遅延化に関しては 2.4 節で詳しく述べている．

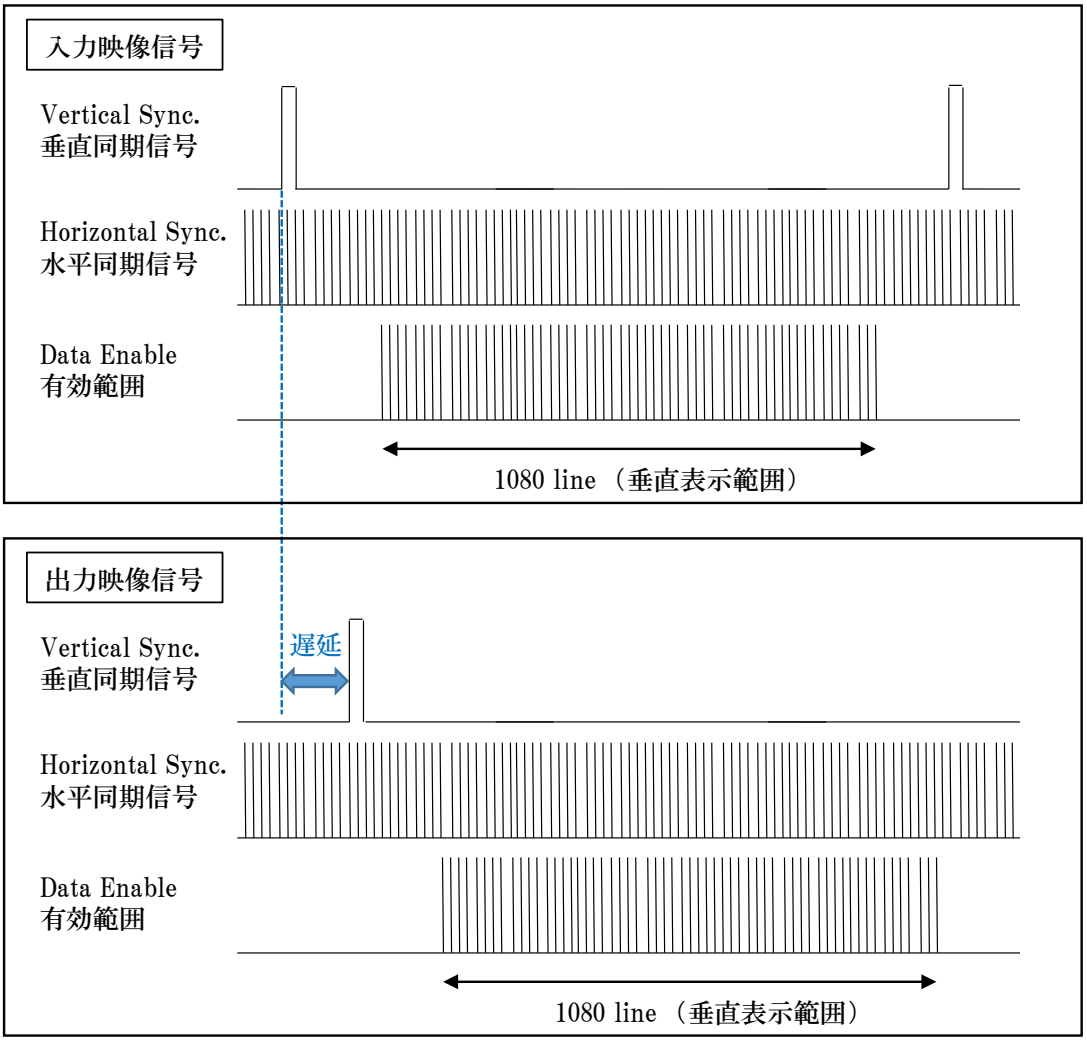


図 2.2.5：低遅延リアルタイム処理（ 制御信号のみ表示 ）

2.2.3　並列演算の必要性

2.2.2 項では，映像表示システムにおける低遅延リアルタイム処理の枠組みと，それを実現する上で必要となるアルゴリズム上の要件について述べた．本項では，解像度の高い映像表示システムにおいて，低遅延リアルタイム処理を実現するには ASIC（Application Specific Integrated Circuit）や FPGA（Field-Programmable Gate Array）といった LSI（Large-Scale Integrated circuit）による実現が有効である点について説明する．まず，一般的な例として Full HD として知られる 1080p 60Hz の信号をリアルタイム演算処理で実現することを例に考えると，ブランク期間を含めた垂直解像度 1125 line と水平解像度 2200 pixel，及び，リフレッシュレート 60 Hz から，1 画素の伝送レートを現す Pixel Clock の周波数は：

$$2200 \text{ pixel} \times 1125 \text{ line} \times 60 \text{ Hz} = 148.5 \text{ MHz}$$

となる．つまり，ブランク期間を無視すれば，148.5 MHz のスピードで出力画素を送受信し続けなければならないことになる．例えばこれを 4.0 GHz 程度の CPU で実現しようとすると，マルチコア等を考慮しなければ，単純計算で約 27 回（Clocks）程度の演算で 1 画素の入出力をしなければいけないことになる．この演算量は 5×5 サイズ程度のフィルター演算でさえも厳しいといわざるを得ないレベルの演算性能である．また，仮に 4K 解像度（3840 × 2160）になった場合，ブランク期間を無視したとしても：

$$3840 \text{ pixel} \times 2160 \text{ line} \times 60 \text{ Hz} = 497.7 \text{ MHz}$$

最低 497.7MHz のスループットが必要となり，同じく 4.0 GHz の CPU の場合は約 8 回（Clocks）程度の演算しかできないことになり，非常に簡易な演算でさえも実現できる可能性が低いことがわかる. このように, 必要演算性能に対する高解像度化の影響は甚大で, CPU のような中央演算型の処理だけではリアルタイム映像処理の実現は解像度が低く, かつ，簡易な処理である場合を除いて困難である．それ故，高解像度映像のリアルタイム処理を実現するには，GPU（Graphics Processing Unit）や ASIC, FPGA のような並列演算処理に特化したハードウェアデバイスが必要不可欠となる．実際に，従来の 4K システム等は内部で 2K 演算回路を 4 つ並列化することで実現されているのが一般的で, その結果, 物理的に制限されたクロック周波数でも高い解像度の演算や表示ができている．なお，GPU か ASIC，FPGA のような LSI かといった選択については対象とする演算内容の規模に依存する部分ではあるが，本研究での超解像システムでは，LSI の使用を想定している．計算規模に関する詳細理由は 2.4 節で述べるが，専用ハードウェアロジックを自由に設計できる ASIC や FPGA のほうが高解像度に対する適応性が高く，かつ，OS を必要としない高い安定性も確保できるといった利点がある．

2.3. 機械学習による辞書型超解像

前節では，映像表示システムにおける「超解像」システムの枠組みと，低遅延でかつリアルタイム処理を実現する上で必要なアルゴリズム要件について説明した.本節以降では，上記要件に基づいた視点から，従来の超解像技術を見つめ，課題や利点を述べていく.

2.3.1 Example-based Super-resolution

C.Y. Yang らの報告 [2-2] の例にも在る通り，1 枚超解像（ Single Image Super Resolution ）に関する歴史は古く，実に様々な手法が存在する．例えば，従来の学習を要しない方法としては，斜め線のジャギーを発生させないように考慮した補間方式に基づく手法や，ぼけたエッジを際立たせることに注力したエッジ方式などが存在している．これらの技術は，一般には何らかの先見情報に基づいてアルゴリズム設計者である人間がルールを設定し，そのルールに従って処理を施すというものであった.

一方で，近年特に注目が高まっている学習型超解像（ Learning based Super Resolution ）では，機械学習（ Machine Learning ）技術によって蓄えられたパターンを先見情報として利用している．それ故，設計者の主観に依存しておらず，人間が設計するよりも緻密で複雑なパターンにまで対応できる点が特徴となっている.

学習型超解像の枠組みを初期に提案した論文として W.T. Freeman らが 2002 年に ECCV（ European Conference on Computer Vision ）で発表した Example-based Super-resolution（ 国内では事例参照型超解像とも呼ばれる ）[2-3] がある.

事例参照型超解像では，対となる低解像度画像と高解像度画像から抽出された対となるパッチ画像集合，つまり，局所的な低解像度成分と高解像度成分がペアになった辞書とも呼ばれるデータベースを形成する．復元時には，その辞書を参照し，対象と合うパターンを見つけたらその高周波成分を類似度に応じて加重平均して適用するといった手法である [2-3]．本手法を実践すると，入力画像には含まれていなかった高周波成分が付加され，画像の細部が復元される.

ここで，事例参照型超解像を低遅延リアルタイム処理という観点で眺めてみると，まず，パッチベースで処理できるという利点が挙げられる．これはパッチ処理の計算フローさえ実現できれば，任意の画像サイズにも対応できることを意味すると同時に，画面全体を見る必要が無いため，少ない遅延で実現できる可能性があることを意味している．また，処理が one-pass であり，基本的なフローに反復演算が含まれておらず，固定時間で処理を完結させることができるという利点を持つ．これらの特徴は一定のペースで入出力ストリームのスループットを維持しなければならない低遅延リアルタイム処理を実現する上で有利な特徴であるといえる.

しかしながら，田口らの報告 [2-4] にもある通り，あらゆる入力画像に対して高い画質を得るには，記憶する事例の数を莫大に増やさねばならないという本質的な課題が事例参照型超解像にはある．これは容易に予想できることであるが，パッチサイズに依存して，入力が取りうる値のバリエーションは膨大な数になる．いわば，パッチサイズで表現可能

な全ての画像が対象といっても過言ではなく，それらのパターンを全て格納するには非常に大規模なメモリが必要になる．これは例えば，僅か 4×4 ピクセルのサイズをもった画像パッチでさえも，画像情報が RGB 24bit の場合，2^{30} ＝約 10 億通り以上のパターンが存在することからも明らかである．また，さらに問題なのは，それら全てのパターンとの比較を入力パッチに対して実施し，その類似度を評価しなければならない点である．

結論として，W.T. Freeman らの手法は，予め用意した辞書ともいえるデータベースを検索して入力事例と類似な事象を参照し，その類似度によって結果を推定するというものであり，人間の直観にとても近い学習型超解像手法であるといえる．なぜならば，辞書に含まれる情報は人間が見る画像そのものであるからである．それらのパターンをそのまま保持するという点が分かりやすい反面，次項以降で紹介していく手法と比較すると，最適化という部分では冗長であるといえる．理由については次項から詳しく説明していく．

2.3.2 ScSR (Sparse-coding Super Resolution)

W.T. Freeman の事例参照型超解像が，辞書を用いた初期の学習型超解像であったのに対し，J. Yang らが 2008 年に画像認識系の国際学会である CVPR（Conference on Computer Vision and Pattern Recognition ）で発表したのが当時のトレンドでもあったスパースコーディング（ スパース信号表現とも呼ばれる ）を取り入れた ScSR（Sparse-coding Super Resolution) [2-5] であった．ここで，スパース（ sparse ）とは ”疎らな” と和訳されることが多いが，要するに対象信号（ 画像処理の場合は対象画像 ）を非常に少ない僅かな信号（ 基底ベクトル ）で表す手法のことを指し，数式上では式（2.3.1 ）で表現される．

$$\min \|\boldsymbol{\alpha}\|_0 \ \ s.t.\ \boldsymbol{y} = \boldsymbol{D}\boldsymbol{\alpha} \qquad (2.3.1)$$

ここで，$\boldsymbol{y}$ は対象画像パッチを表す列ベクトル，$\boldsymbol{D}$ は $\boldsymbol{y}$ を構成するための基底ベクトルを含んだ辞書，そして $\boldsymbol{\alpha}$ は辞書内の僅かな基底ベクトルを選択するためのスパースな係数行列を示している．また，$\min \|\boldsymbol{\alpha}\|_0$ は，$\boldsymbol{\alpha}$ の L0 ノルム（ すなわち，非ゼロの要素数 ）を最小化することを意味しているが，一般に上記をそのまま解くことは難しく，L0 ノルムを L1 ノルムへと制約緩和した式（2.3.2 ）の近似解で代替される．

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1, \ \lambda > 0 \qquad (2.3.2)$$

ただし，λ は正則化項の重みを決定付けるパラメータである．

J. Yang らは文献 [2-5] で，上記スパースコーディングの基本式を超解像に適用することを提案した．その手法では，高解像度画像と低解像度画像で対となる辞書を作成し，その辞書を 1 つのスパース係数で共有することで超解像システムを実現している．具体的には，N 次元の高解像度画像 X^h と，それに対応する M 次元の低解像度画像 Y^l が与えられ

たとき，それらの画像をスパース表現可能な辞書 $\{ D^h, D^l \}$ を，式（2.3.3）の最適化問題として求め，スパース表現可能な辞書を作成した．

$$\min_{\{\boldsymbol{D_l},\boldsymbol{D_h},\boldsymbol{Z}\}} \|\boldsymbol{X}_c - \boldsymbol{D}_c\boldsymbol{Z}\|_2^2 + \hat{\lambda}\|\boldsymbol{Z}\|_1, \quad \hat{\lambda} > 0$$

$$\boldsymbol{X}_c = \begin{bmatrix} X^h/\sqrt{N} \\ Y^l/\sqrt{M} \end{bmatrix}, \boldsymbol{D}_c = \begin{bmatrix} D^h/\sqrt{N} \\ D^l/\sqrt{M} \end{bmatrix} \qquad (2.3.3)$$

上式によって最適化された辞書 $\{ D^h, D^l \}$ は Z という共通のスパースな係数行列に対応した一対の辞書となる．従って，入力を低解像度画像の辞書 D^l でスパース表現した際の係数をそのまま対となる高解像度画像の辞書 D^h に適用し，高解像度成分を復元することが可能となる．実際，文献 [2-5] にも示されている通り ScSR の復元結果は Bicubic 補間と比較して細部のテクスチャが復元されている様子が分かる．これらのテクスチャは式（2.3.3）における辞書 D^h の基底ベクトルを組み合わせて形成されたものである．また，文献 [2-5] には高解像度側の辞書 D^h を可視化した図も掲載されているが，スパース制約によって生成された辞書が，従来の 2 次元フーリエ変換で見られるようなコサイン直交基底とは全く異なる様相を呈していることが分かる．この違いはスパース性が担保できるかどうかにある．つまり，任意の信号の分解はフーリエ級数展開によっても表現可能であるが, 信号を再現する上で必要となる基底関数の個数は, 必ずしも少数になるとは限らない．一方で，スパース制約を最適化されるべく学習された基底関数であれば，従来の直交系基底ベクトルに比べて非常に少ない基底ベクトルの組み合わせで近似的に表現できる．なぜならば，式（2.3.3）で示した通り，出来るだけそうなる辞書が得られるように学習されているからである．

以上が ScSR の概略であるが，ScSR を低遅延リアルタイム処理の実現という観点で評価すると，利点と欠点がやはり存在する．まず，利点としては，処理がパッチベースであることが挙げられる. これは低遅延を実現する上での必須条件といっても良い. それから，スパースコーディングによって辞書が圧縮されている点も非常に優れた点である．W.T. Freeman が提唱した事例参照型超解像では, 各事例をいわばそのままの状態で保持する必要があったが，ScSR では，推定すべき出力成分を基底ベクトルに分解し，再構築するという手段をとっているため，全パターンの事例を保持する必要が無く，情報が圧縮されている．例えば，文献 [2-5] で示された高周波成分を形成する辞書に含まれる atom とも呼ばれる基底ベクトルの数は，画像パッチのサイズが 9×9 ピクセルの大きさに対し，たかだか 1024 個や 512 個といった個数であった．これらの個数は 2.3.1 項でも触れた該当パッチサイズの画像が表現可能な画像パターンのバリエーションから比較すると極めて少ない数である．また，選ばれた基底ベクトルの数がスパース（ 疎ら ）となる点も利点といえる．これはハードウェアの復元回路として，ほんの数個の基底ベクトルを組み合わせる回路さえ実現できれば，ある程度の精度で近似解が期待できることを意味している．

一方で, ScSR にも欠点が存在する. それはスパース係数の算出コストである. 式（2.3.2）は L1 ノルムによって簡略化された式とはいえ, 解析解を得ることは出来ない. そのため,

OMP（ Orthogonal Matting Pursuit ）法 [2-6] などの反復計算を用いて数値解を得る必要がある．ここで，OMP 法は表現したい信号の残差成分を表現する上で最も有効となる内積値の高い基底ベクトルを一つずつ選択しながら残差成分を更新し，精度を高めていくという反復手法であるが，スループットを維持するために一定時間内に処理を完結しなければならないリアルタイム処理では反復演算は好ましくない．勿論，固定の反復回数で打ち切るといった工夫も考えられるが，基本的には性能が落ちることになる．また，有効な基底ベクトルを探す際に，各基底ベクトルとの内積を計算しなければならない点も計算コストの観点で問題である．探索しなければならない量は W.T. Freeman の事例参照型超解像ほどではないが，辞書に含まれた全ての基底ベクトルと対象信号との内積計算を反復演算毎に行わなければならないのは非常に計算コストが高い処理であり，特に，全ての処理をパイプライン化して並べなくてはならないハードウェア処理としては大規模かつ複雑な処理にならざるを得ない．

このように W.T. Freeman が事例参照型超解像として開始した辞書型超解像は ScSR の出現によって，より高度な最適化を果たすことが出来た．ScSR は機械学習によって生成された辞書によって実現されており，最早，人間の手では生み出せない高度な最適化が成されているといえる．しかしながら，その辞書を活用するという部分では機械学習に頼っておらず，あくまで人間から見て分かりやすい構造をしていた．その結果，復元処理の実現面での計算量増大という課題に繋がったと考えられる．この部分に対する解決が End-to-End と呼ばれる機械学習によって克服されるのだが，それについては次節で詳しく述べる．

2.4　畳み込みニューラルネットによる超解像（SRCNN）

前節では，学習型超解像の黎明期ともいえる辞書型の超解像について，低遅延リアルタイム処理の実現という立場から説明した．振り返れば，W.T. Freeman の事例参照型学習では機械学習という枠組みを事例の蓄積に用い，J. Yang らは効率的な辞書生成に用いていた. それに対し, 本節で説明する畳み込みニューラルネットによる超解像（SRCNN）では，辞書という概念を捨て，機械学習を低解像度品質の画像から高解像度品質の画像へ変換する手法そのものに適用している点が異なる．言うなれば，超解像という目的に対してよりダイレクトに最適化された学習型超解像手法であるといえる．

2.4.1　システム構成

C. Dong らが 2014 年に ECCV で発表した SRCNN (Super Resolution via Convolutional Neural Network) [2-1][2-7] は図 2.4.1 で示したとおり, 3 層の畳み込みニューラルネットで構成される．ここで，入力画像はバイキュービック補間で拡大済みの画像であり，真に低解像度画像ではないという意味から，入力を低解像度品質の画像，出力を高解像度品質の画像と本論文では表現する．SRCNN を構成する各層の役割を端的に述べれば，第 1 層では入力画像に含まれる情報（特徴）を抽出し，第 2 層ではそれを非線形変換し，第 3 層ではそれらの情報を使って信号を再構築するというものであるが，本構成は SRCNN の考え方や今後の計算コスト等の課題を理解する上で重要であるため，ここではより詳しく，各層で行われる処理の詳細について述べていく．

まず，特徴抽出とされる第 1 層では式（2.4.1）に基づく計算が行われる．

$$F_1(\boldsymbol{Y}) = \mathrm{ReLU}(W_1 * \boldsymbol{Y} + B_1) \qquad (2.4.1)$$

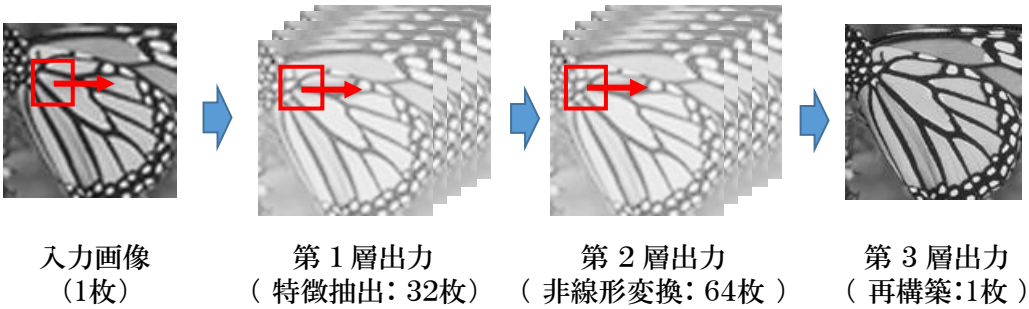


図 2.4.1: SRCNN の構成

ここで ＊ は畳み込み演算を表しており，入力画像と同じ面積の画像が出力される．具体的には, 上記の計算は 9×9 ($f_1 = 9$) サイズの畳み込みフィルターを 64 種類 ($n_1 = 64$) 適用し，64 枚のフィルター処理画像を生成することに等しい．また，重み行列 W_1のサイズは $1\times f_1 \times f_1 \times n_1$ であり, B_1 は 1 層の出力であるn_1次元のバイアス成分となる. また，ReLU (Rectified Linear Unit) は式（2.4.2）で定義される活性化関数である．

$$\mathrm{ReLU}(\boldsymbol{x}) = \max(0, \boldsymbol{x}) \qquad (2.4.2)$$

ここで，活性化関数とは各層の出力と入力を接続する際に挿入される関数であり，ニューラルネットの入出力特性に対して，非線形性を付加する役割を担う．仮に，非線形な活性化関数が無い場合は多層のニューラルネット構造は単なる線形写像に置き換えられてしまうため，高度で複雑な変換には対応できなくなる．つまり，活性化関数の非線形性は，多層構造を単なる線形写像ではなく，より高度な入出力関係性を学習・推定するために必要不可欠な要素であるといえる．

次に，非線形性変換と表現された第 2 層では，第 1 層と同様に，式（2.4.3）によって計算が行われる．

$$F_2(\boldsymbol{Y}) = \mathrm{ReLU}(\, W_2 * F_1(\boldsymbol{Y}) + B_2) \qquad (2.4.3)$$

ここで，重み行列 W_2のサイズは 入力次元 n_1，出力次元 n_2，畳み込みフィルターのサイズ f_2 から，$n_1 \times f_2 \times f_2 \times n_2$ と定まるが，これらのパラメータによってコストと性能が左右されるため，J. Yang らはハイコストだが性能を重視した SRCNN 9-5-5 モデル : $(f_2, n_2) = (5, 32)$ と, 若干性能は低下するがローコストな SRCNN 9-1-5 モデル : $(f_2, n_2) = (1, 32)$ の 2 パターンを提案した．いずれにせよ，第 2 層での処理は 64 枚の画像から 32 枚の画像生成を意味し，9-5-5 の場合は，5×5 フィルターを用いてそれを実現し，9-1-5 の場合は，1×1 フィルターで実現をすることになる．

そして，最後に再構成層と表現された第 3 層では，式(2.4.4) で表される畳み込み演算によって，１枚の画像が生成される．

$$F(\boldsymbol{Y}) = \, W_3 * F_2(\boldsymbol{Y}) + B_3 \qquad (2.4.4)$$

ただし，重み行列 W_3 のサイズは $n_2 \times f_3 \times f_3 \times 1$ であり，畳み込みフィルターのサイズは $f_3 = 5$ である．つまり，5×5 サイズの畳み込みフィルターを 32 種類使って，32 枚の画像から 1 枚の画像を合成するといった処理が行われる．なお，最終段は値を予測する回帰（Regression）問題なので，活性化関数は省かれている．

2.4.2 End-to-End 学習による復元性能

2.4.1 項で説明した SRCNN の出力は全て各層のパラメータである $\Theta = \{ W_1, W_2, W_3, B_1, B_2, B_3 \}$ に依存する．従って，SRCNN が超解像として正常に機能するには，これらのパラメータを適切に設定する必要がある．C. Dong らはペアとなった低解像度品質画像 $\boldsymbol{Y}$ と高解像度品質画像 $\boldsymbol{X}$ を複数枚用意し，それらを入力，及び，理想出力（教師データ）と捉えることで，誤差逆伝播法（Back propagation）により上記パラメータを最適化する手法を提案した．その際の損失関数は式（2.4.5）で与えられる．

$$L(\Theta) = \frac{1}{n}\sum_{i=1}^{n} \| F(Y_i; \Theta) - X_i \|^2 \qquad (2.4.5)$$

誤差逆伝播法では，出力側から入力側に向かって，上記の損失関数を軽減する勾配方向にパラメータ調整が反復的に行われる．従って，この反復を繰り返すたびに式（2.4.5）の損失関数の値は減少していき，結果的にニューラルネットの出力である $F(\boldsymbol{Y})$ は，理想的な出力である $\boldsymbol{X}$ に可能な限り近づけられる．実際，文献 [2-7] にも示されている通り，SRCNN は学習を繰り返すことで性能が向上し，ScSR を含む当時の state-of-the-art の手法を上回った復元性能が確認されている．

このように，SRCNN は入力画像である低解像度品質の画像と，理想的な出力画像である高解像度品質の画像を学習用データセットとして事前に用意し，それらをネットワークに与えるだけで，新たな低解像度品質の画像に対しても高い精度で理想的な状態に近づけることが出来る．このようなアプローチは末端情報だけで内部をブラックボックスとして扱える点から End-to-End 学習といわれている．

End-to-End 学習の概念は現代のニューラルネット型超解像における基本的な思想となってきているが，従来の辞書型超解像とは全く異なる概念であることを改めて述べておく．つまり，W.T. Freeman の事例参照型超解像や，J. Yang らの ScSR で代表される辞書型超解像では，あくまで超解像システムの一部を機械学習に任せていたが，End-to-End 学習である C. Dong らの SRCNN では，ニューラルネットの構成やフィルターサイズなどのハイパーパラメータは除くものの，超解像システム全体を機械学習に委ねたという点が大きく異なる．この点についてさらに補足すれば，End-to-End 学習によって形成された SRCNN には，もはや辞書という概念は存在しておらず，代わりに各層の畳み込みフィルターを実現するための重み行列が先見情報を蓄積し，かつ，反映する役割を担っている．

例えば，文献 [2-7] には End-to-End の学習の結果得られた SRCNN 第 1 層の 9×9 サイズのフィルターを可視化したものが図で示されているが，高周波成分を抽出するためのフィルターが自然と形成されていることがわかる．これらの特徴抽出フィルターは従来であれば人間の手で設計されていたものであったが，機械学習に委ねることによって，より複雑な形状に対応した最適化が行われ，当時の従来性能を上回る成果が得られたと解釈できる．

また，SRCNN の更なる利点として，従来の復元に比べて計算量の少ないフィードフォワード演算のみで高精度の推定が実現できている点が挙げられる．C. Dong らが行った PC

による処理時間の比較が文献 [2-7] に掲載されているが，SRCNN は従来の ScSR と比較しても少ない処理時間で推定を実現している．ここで，誤差逆伝播法によって調整されるのはあくまでネットワーク内パラメータ（重み行列値）であり，中間次元数やフィルターサイズなどのハイパーパラメータのような演算量を決めるネットワーク構成に影響しない．つまり，SRCNN は ScSR と同等の演算量で構成されるネットワークが与えられたとしても，最適化によって高い復元精度にまで到達出来ることを意味する．以上の理由から，超解像という課題に対し，辞書型超解像よりもさらに効率の高い機械学習による最適化が SRCNN では成されていると捉えることができる．

2.4.3 低遅延リアルタイム処理への課題

本項では低遅延リアルタイム処理という観点から SRCNN の課題について述べていく．既に 2.4.2 項でも述べた通り，フィードフォワード演算によって構成される SRCNN は，演算処理が固定長で完了できるため，一定のスループットを保持するリアルタイム処理には適した処理といえる．しかしながら，SRCNN を映像の低遅延リアルタイム処理に適用するには，処理遅延や処理コストという観点で課題が存在する．

例えば，一般に公開・検討されている画像処理アルゴリズムの多くは PC 上でのソフトウェア処理を前提とした処理を想定している場合が多いが，SRCNN も同様である．PC 上のソフトウェア処理では，画像ファイルがハードディスクに保存されており，画像ファイルのデータを DRAM 等のメモリに展開して画像処理が行われる．具体的に SRCNN を例に考えると，3 層 CNN の各中間層では，第 1 層で 64 枚の画像を，第 2 層で 32 枚の画像をそれぞれ出力し，それらの画像データ（サイズは入力と同等）を次層の入力として計算を行わなければならない．本手法はメモリを大規模に利用することで，トータルの計算量を削減することが可能となるが，例えば 1 画面の画像を処理するためには 1 画面分の情報を予め知る必要があり, 2.2.2 項で説明したリアルタイム処理の枠組みで実現するためにはフレーム遅延が免れない．

この問題に対し，S.Y. Kim らは入力画像を予め小さなタイルに分割して，GPU で並列処理させることで, リアルタイム処理を実現する検討を行った [2-8]. この結果を見ると，SRCNN を GPU 処理に最適化することで，格段に処理時間を削減している．しかしながら，2560×2048 の画像サイズの処理時間は GPU を用いても 539.90 msec も掛かっている．勿論，本数値はマシンスペックに依存するところではあるが，60 fps 時における 1 フレーム処理時間が 16.6msec であることや, 4K 解像度は更に大きな画像（ 3840×2160 ）であることを考えると，文献 [2-8] の結果からは 4K や 8K といった高解像度映像のリアルタイム処理は達成できていないことが分かる．

一方で，SRCNN の実行に必要となる計算量は，マルチコアなど内部演算の効率化を無視すれば，フィルターサイズと各層の入出力次元数から簡易的に計算が出来る．例えば，SRCNN 9-5-5 の場合，第一層を考えると，輝度成分のみを保持した 1 次元の 4K 画像に対して，64 次元の処理画像を 64 種類の 9×9 フィルターで計算することから：

$$\text{SRCNN 第 1 層}：9 \times 9 \times 3840 \times 2160 \times 1 \times 64 = 4.300\text{E} + 10 \qquad (2.4.6)$$

約 4.300E+ 10 回（430 億回 ）の積和計算が必要なことが分かる．同様に，第 2 層，第 3 層を計算すると：

$$\text{SRCNN 第 2 層：} 5 \times 5 \times 3840 \times 2160 \times 64 \times 32 = 4.247\text{E} + 11 \quad (2.4.7)$$

$$\text{SRCNN 第 3 層：} 5 \times 5 \times 3840 \times 2160 \times 32 \times 1 = 6.636\text{E} + 09 \quad (2.4.8)$$

となり，合計 4.743E+11 回（ 4743 億回 ）の積和演算が 4K 画像 1 枚辺り必要となることがわかる．なお，実際にはこれを 60 fps で駆動しなければならないため，結果として約 28 TFLOPS の演算が必要となる．この数値は 2018 年現在において，非常に高価な GPU を以ってしても実現性困難な計算量である．また，同様の計算から約 4TFLOPS の計算で実現できる SRCNN 9-1-5 モデルであれば，ハイエンドのグラフィックボードである NVIDIA Quadro シリーズの M6000 [2-9] で実現できる可能性もあるが，本研究の目的である表示機器への超解像搭載という観点からみるとコスト面，電力面の理由からも非現実的である．

2.4.4 ハードウェア実装コストの見積もり

2.4.3 項ではソフトウェア実装を前提とした処理遅延と計算コストに関する課題を述べたが，やはり 4K や 8K 等の高解像度映像に対するリアルタイム処理はソフトウェア実装では未だ厳しく，ASIC や FPGA で代表される LSI を用いた専用ハードウェアロジックによるパイプライン型並列演算が有力な候補となる．しかしながら，単純に SRCNN のハードウェア実装を考えると，実装コストにおける課題がある．

まず，最も少ない遅延時間を実現するためのシンプルな考え方としては，各画素の結果を出力する回路構成を検討することである．図 2.4.2 は SRCNN 9-5-5 モデル，すなわち，第 1 層のフィルターサイズが 9×9，第 2 層のフィルターサイズが 5×5，第 3 層のフィル

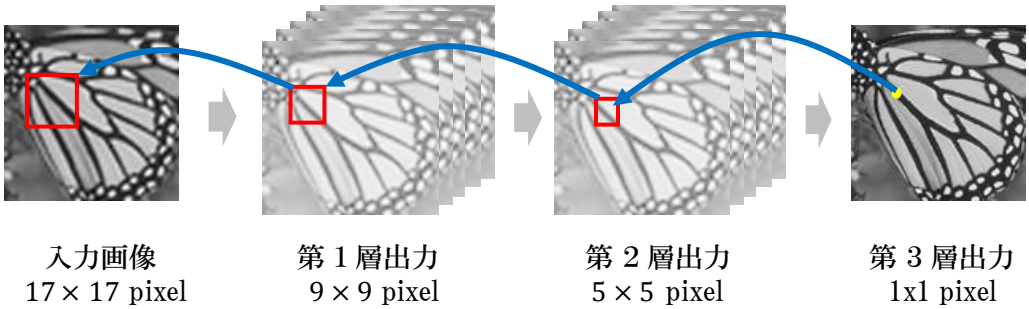


図 2.4.2：受容野（Receptive Filed）の拡がり

ターサイズが 5×5 の場合において，出力の 1×1 pixel の画素値を決定するのに必要となる領域を図示したものである．受容野の範囲は出力層から入力層に向かって層を重ねるごとに広がって行き，今回のケースでは 17 × 17 pixel の受容野（Receptive Field）を持つことが分かる．つまり，第 1 層のハードウェアの演算コストとしては，入力範囲が 17 × 17 pixel（ 1 次元 ） で，出力範囲が 9×9（64 次元 ）となることから，

SRCNN 第 1 層の積和演算量：9×9×1×64×9×9 = 419,904 （2.4.9）

約 41 万ステップの演算回路が必要であり，計算に必要な領域のメモリ量としては，

SRCNN 第 1 層のメモリ量：17×17×1 = 289 （2.4.10）

289 pixel の中間データ（実際には 32 bit 浮動少数点等の数値）を確保する分が必要となる．同様にして，第 2 層，第 3 層を計算すれば，

SRCNN 第 2 層の積和演算量：5×5×64×32×5×5 = 1,280,000 （2.4.11）

SRCNN 第 3 層の積和演算量：5×5×32×1×1×1 = 800 （2.4.12）

SRCNN 第 2 層のメモリ量：9×9×64 = 5,184 （2.4.13）

SRCNN 第 3 層のメモリ量：5×5×32= 800 （2.4.14）

となり，合計 1,700,704 ステップの積和演算と，6273 pixel 分に相当するメモリが中間演算結果の格納として必要なことがわかる．つまり，1.1.4 項でも述べたが，結論としては SRCNN 9-5-5 の場合は約 170 万回，SRCNN 9-1-5 の場合は約 18 万回の積和演算が必要であり，これらの計算量は非常に大きい．例として，比較的大規模で 4K 解像度の映像処理にも対応できる FPGA として知られる Xilinx 製の Kintex UltraScale [2-10] の最高峰である KU115 でも高精度な演算が可能な DSP スライスの個数は 5,520 個である．また，各 DSP に備わる掛け算回路は 27 bit ×18bit までしか対応していない．つまり，DSP だけで上記計算量を実現するのは不可能である． 勿論，DSP で足りない部分は System Logics Cells で補えるが，それでも 145 万回路の Logic Cell で SRCNN 9-5-5 実装はまず不可能であるといえる．また，積和演算が実際には 32bit 程度の浮動少数点演算である場合は，Xilinx のホワイトペーパー [2-11] にもある通り，ハードウェア消費リソースは固定少数点演算の場合に比べて 5 倍近く増加することから，実際には浮動少数点演算による SRCNN 9-1-5 モデルの実装もできないという結論になる．いずれにせよ SRCNN のハードウェア実装は未だハイコストであり，課題があるという状況に変わりはないのが現状となる．

2.5　SRCNN 以降のニューラルネット型超解像

SRCNN は Deep Learning で確立された End-to-End 学習を超解像の分野に適用させることで高い性能が得られることを初めて世に知らしめた手法であり，現在の Deep Learning 型超解像の基礎を形成したともいえる．本節では，超解像分野で SRCNN 以降に公開された幾つかの論文手法についてまとめ，ポスト SRCNN の技術動向を述べると共に，本研究の主目的である低遅延リアルタイム向け超解像技術の実現に向けた課題や注目技術について述べていく．

2.5.1　深層化による性能向上

Deep learning 研究が盛んになり始めた 2015 年頃，“The Deeper, the Better”という言葉をよく耳にした．つまり，より深層化した方が結果も良くなる，という意味合いである．J. Kim らが 2016 年の CVPR で発表した VDSR（Very Deep Super Resolution）[2-12]も正にその考え方に準じるもので，従来 3 層であった SRCNN をさらに多層化し，実際には 20 層まで拡張して復元性能を上げたものであった．

また，VDSR のアーキテクチャの特徴としては，一般画像認識の分野で He らが提案した ResNet（Residual Net）[2-13] の構成が取られている点が挙げられる．ResNet では入力の画像情報が分岐して前方にそのまま連結されるため，結果としてニューラルネットで推定すべき部分は教師データである高解像度品質画像と入力である低解像度品質画像の差分（Residual）のみでよくなる．このような ResNet 構成を選択することで，学習対象は未知である失われた高周波成分となり，入力成分に含まれた低周波成分まで再現しなくて済むようになる．さらに，入力信号の伝達性が高まることによって深層化で起きやすいとされる勾配消失などの問題を抑制し，学習パラメータの収束性が高まるといった効果も知られている．

文献 [2-13] には VDSR が SRCNN と比べてより高い復元性能を持っていることが述べられているが，それに加えて，同一パラメータで複数の拡大率にも対応できている点も述べられている．VDSR が複数の解像度でも高い再現性を実現出来ている理由は，予め学習データに複数の拡大率の教師データを含めているからであるが，それでも性能が高まっている理由としては, 20 層もの多層化によってネットワークの自由度が大幅に高まった結果, より多くの先見情報を蓄積できるようになったと捉えることができる．

しかしながら，問題は実現コストにある．VDSR のネットワーク構成パラメータとしては，各層のフィルターサイズが 3×3，中間層の次元数が 64 に設定されている．フィルターサイズが SRCNN に比べて比較的小さいのは計算コストを抑えて成るべく深層化を優先したいという目的と見られるが, それでも 20 層重ねたときに得られる最終的な受容野は，41×41 pixel まで拡がり，SRCNN の 13 × 13 pixel よりも広い．そして，2.4.4 項と同様に 1 pixel の出力値を計算する上で必要となるハードウェア構成の見積もりを計算すると，表 2.5.1 の結果となり，結論として約 3.3 億 step の積和演算と，約 68 万 pixel のメモリが必要となることが分かる．なお，本数値を SRCNN 9-5-5 モデルと比較すれば，約

198 倍であり，明らかに 2.4.4 項で説明した現状のデバイス規模からみても非現実的なコストであることがわかる．

以上の通り，VDSR は深層化によって SRCNN の性能を更に高めた End-to-End 学習型の超解像における代表的な手法といえるが，実装のコストが非常に大きく，本研究の目的であるディスプレイ上での低遅延リアルタイム超解像の実現という視点からは方向性が外れたアプローチであると認識できる．

表 2.5.1：VDSR の実装コスト

Layer	Filter Size	Input Dim	Output Dim	Receptive Field	メモリ [pixel]	積和演算数 [step]
20	3	64	1	3	576	576
19	3	64	64	5	1,600	331,776
18	3	64	64	7	3,136	921,600
17	3	64	64	9	5,184	1,806,336
16	3	64	64	11	7,744	2,985,984
15	3	64	64	13	10,816	4,460,544
14	3	64	64	15	14,400	6,230,016
13	3	64	64	17	18,496	8,294,400
12	3	64	64	19	23,104	10,653,696
11	3	64	64	21	28,224	13,307,904
10	3	64	64	23	33,856	16,257,024
9	3	64	64	25	40,000	19,501,056
8	3	64	64	27	46,656	23,040,000
7	3	64	64	29	53,824	26,873,856
6	3	64	64	31	61,504	31,002,624
5	3	64	64	33	69,696	35,426,304
4	3	64	64	35	78,400	40,144,896
3	3	64	64	37	87,616	45,158,400
2	3	64	64	39	97,344	50,466,816
1	3	1	64	41	1,681	876,096
				Total	683,857	337,739,904

2.5.2　拡大補間画素位置に基づく性能向上

VDSR が深層化に伴う性能向上手法であるのに対し，それとは全く異なるアプローチによる性能向上提案として，ESPCN（Efficient sub-pixel convolutional neural network）と呼ばれる手法が VDSR と同じ 2016 年の CVPR で Shi らによって報告された [2-14]．図 2.5.1 は 2 倍拡大を例にした ESPCN の構成図である．2 倍拡大の場合は入力画像が縦横 2 倍，すなわち，4 倍の解像度になるため入力画像 1 画素は拡大画像の 4 個のサブピクセルに相当する．そこで低解像度の入力画像から 4 つの各サブピクセル位置に対応した画

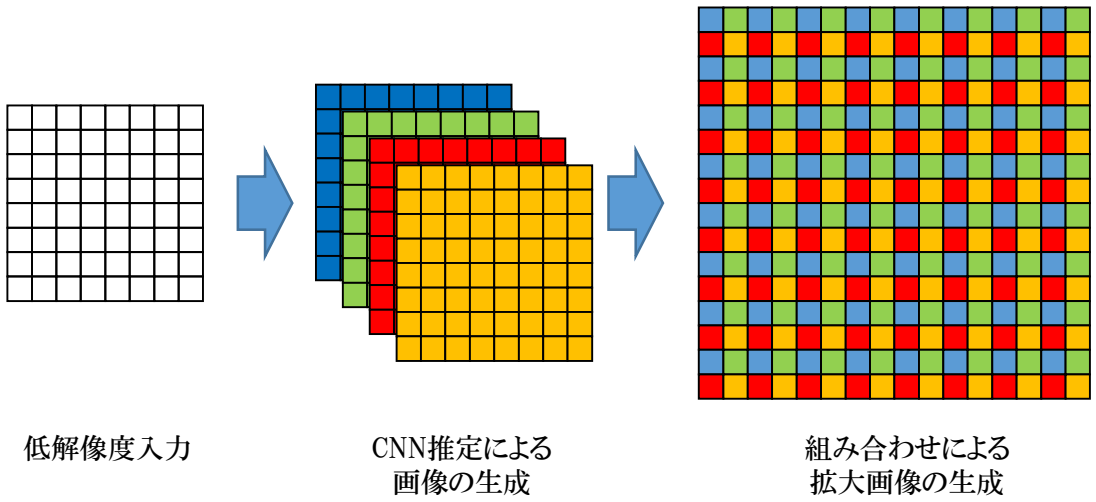


図 2.5.1：拡大補間画素位置別のＣＮＮ推定

像を CNN によって生成し，それらの出力を組み合わせることで高解像度画像を得るといった手法である．原理上，入力画像は真に低解像度の画像が必要となる点から，1.2 節で述べた解像度変換を行わないという本研究の対象からは外れるが，文献［2-14］の結果によると，従来の SRCNN と比較して復元性能が向上していることが確認されている．

一方で，同時期の 2016 年に発行された電子情報通信学会の速報にて，大谷らが Shi らとほぼ同様の提案を報告していた［2-15］．大谷らはバイキュービック拡大画像が 4 種類の異なる重みによって計算される点に着目し，従来の SRCNN ではこれらの違いを考慮せずに 4 画素に対して同一の CNN で処理を行っているが故にジャギーやリンギングといったアーチファクトが発生しているという仮説を立てた．そして，入力画像に対して独立な 4 つの CNN を並列に接続し，出力された画像を拡大画像の各サブピクセルとして扱うことで，組み合わされた 1 枚の出力画像を得るという手法を提案した．

大谷らの手法は Shi らの手法と本質的には同じであるが，いずれの手法も入力画像としては拡大前の状態を想定している．拡大前のサイズの小さい画像を入力にすることで，ネットワーク規模の削減も見込めるとはいえ，拡大率の 2 乗に応じた並列の CNN を持つ必要があり，処理構成としては複雑で，かつ，計算コストが高い（基本構成は SRCNN と同様であり，計算量も同程度のオーダーが必要）という課題が残る．

2.5.3　Auto-encoder を組み合わせた超解像

CNN で構成された SRCNN には遅延時間や処理コスト等に課題があることは 2.4.3 項で述べた．一方で，SRCNN とは全く異なるニューラルネット構成によってこれらの課題を解決する可能性を示したのが，K. Zeng らが 2017 年に報告した Coupled Deep Auto-encoder（CDA）であった［2-16］．CDA は SRCNN とは全く異なる考え方に基づいて構成されたニューラルネットであるため，以下にその内容について概説する．

まず, CDA を理解する上で必要となる Auto-encoder について述べる. Auto-encoder とは入力信号自身を再現するネットワークであり，その内部に自身をコード化するエンコード部と，エンコードされた値を復元するデコード部がペアとなって備わっている．エンコード・デコードの手法に応じて，実際の Auto-encoder には様々な構成を持ったものが存在するが，K. Zeng らの論文で使用された Auto-encoder としては，以下に述べる構成となっている．

いま，次元数 d のあるパッチ画像 $y_i \in R^d$ が N 個含まれるデータセット $\boldsymbol{Y} = [\mathrm{y}_1, y_2, \ldots, \mathrm{y}_N]$ を用意したとき, 入力信号である y を推定する Auto-encoder を式（2.5.1）で定義する．

$$\begin{aligned} h_i &= f(Wy_i + b) \\ \hat{y}_i &= f(W'h_i + b') \end{aligned} \qquad (2.5.1)$$

ここで, $h_i \in R^n$ は中間層の出力結果であり, エンコードされた信号値を意味する. また, 活性化関数 f は式（2.5.2）のシグモイド関数で定義される．

$$f(z) = \frac{1}{1 + \exp(-z)} \qquad (2.5.2)$$

また，式（2.5.1）の W, W', b, b' は重み行列であり，式（2.5.3）の損失関数 l を最小化するように誤差逆伝播法によって学習される．

$$l = \sum_i \|y_i - \hat{y}_i\|^2 \qquad (2.5.3)$$

以上の構成によって形成された式（2.5.1）のニューラルネットは, 学習によって可能な限り入力信号を復元する Auto-encoder として機能する．

ここで，入力層の出力である h_i は，入力信号 y_i に対して n 種類の（ h_i の次元数 ）の特徴抽出フィルターを掛け合わせて生成した特徴量であると見なすことができ，その n 個の特徴抽出フィルターの集合が $W \in R^{d \times n}$ となる．また，出力層では n 個のパッチ画像と同じサイズである基底ベクトルで構成された $W' \in R^{n \times d}$ の各基底ベクトル（パッチ画像と同様の d 次元を持った W'の各列ベクトル）に h_i の各要素を係数として掛け合わせ，それらを重ね合わせることで入力信号の推定値 $\hat{y}_i$ を合成していると見なせる．

上記 Auto-encoder を, CDA では低解像度品質（LR）用と高解像度品質（HR）用の 2 種用意し，それらを連結することで超解像機能の実現を行っている．具体的には，まず，ステップ 1 として，LR 画像の Auto-encoder を LR 画像のみの学習で生成する．次に，ステップ 2 として，HR 画像の Auto-encoder を HR 画像のみの学習で生成する．そして，ステップ 3 として LR 画像の Auto-encoder で生成された特徴量と HR 画像で生成された特徴量の対応関係を新たな 1 層の全結合 NN で学習させる．つまり，HR 画像の特徴量 h_i^H を，LR 画像の特徴量 h_i^L から式（2.5.4）で推定させる．

$$h_i^H = f(W_2 h_i^L + b_2) \qquad (2.5.4)$$

ただし，ステップ 3 の段階では，LR 画像の encoder 側と，HR 画像の decoder 側の処理を決定する重み行列パラメータは値を固定した状態で式（2.5.4）の最適化を行う. そして, ステップ 4 では最後の fine-tuning として, LR のエンコーダーから HR のデコーダーまでの全ての重み行列パラメータを対象に最適化を再度行い，LR 画像入力に対する HR 画像出力の推定精度が最大限向上するように学習させる．以上が CDA の構成である．なお，K. Zeng らの主張では，上記構成の CDA は，2.3.2 項で説明した ScSR をより一般化したものであるとしている.

ScSR では，予め学習されたペアとなる辞書を用意し，LR 画像である入力信号を LR 側の辞書の単語ともいうべき基底ベクトル（ atom とも呼ばれる ）の組み合わせで表現し，その LR 側の辞書含まれる基底ベクトルを HR 側の辞書に含まれる対応した基底ベクトルに置き換えることで，HR 画像を推定するというものであった．この時，LR 側の推定と HR 側の復元を仲介するスパース係数は共通である．それに対し，CDA では LR 側の encoder で抽出された特徴量（著者らは LR representation と表現）を HR 側の decoder の入力（HR representation）に置き換えている．この構成は ScSR に確かによく似ている．また，LR representation から HR representation への変換が，1 対 1 に対応していた ScSR に比べ，この部分を式（ 2.5.4 ）で示す非線形写像によって決定させる CDA は ScSR の概念を拡張させているという主張も理解できる部分がある．しかしながら，ScSR に含まれていたスパース性については含まれておらず，完全な上位互換ではなくなっている．また，LR と HR のペア型の Auto-encoder を形成するという意義についても，疑問が残る．なぜならば，仮に筆者らが主張する構成が超解像システムを構成する上で重要だというのならば，ステップ 4 の fine-tuning が無くとも高い性能を示すことが可能だが，その部分に関する言及が無い．ステップ 4 の fine-tuning によって LR 側と HR 側の Auto-encoder としての特性は保証されなくなるため，そもそも Auto Encoder を形成する必要性については疑問が残る．なお，文献 [2-16] の処理結果には，復元性能の比較として, SRCNN 9-1-5 との比較も掲載されており, 数値上は若干向上したものの, 文献 [2-7] に示された SRCNN 9-5-5 よりは低い結果であった．つまり，性能面では CDA はそれほど高い向上は果たせていない．しかしながら，文献 [2-16] に記載された処理時間では，CDA は従来の SRCNN 9-1-5 と比較してかなりの軽減効果（約 85% 程度）が報告されている．これらの数値はあくまで PC によるソフトウェア計算による処理時間であって，ハードウェア実装コストでの比較値とは異なるが，パッチベースという点から考えてもハードウェア実装に向いており，処理時間や処理コストの削減という観点で期待を感じるアプローチである.

2.6　本研究で取り組む課題

低遅延リアルタイム処理の実現という観点で近年の機械学習型超解像技術の利点と課題について述べた．本節ではこれまで述べたそれらの課題を踏まえた上で，本研究で検討すべき項目についてまとめる．

2.6.1　低遅延リアルタイム処理に適したアーキテクチャ設計

これまでに述べた通り，超解像の分野においてもディープラーニングをベースとした機械学習が盛んとなり，SRCNN を境に CNN をベースとした超解像の提案が主流となった．しかしながら，4K や 8K などの高解像度ディスプレイでの低遅延リアルタイム処理の実現という観点では，CNN は計算コストが非常に高いという課題がある．一方で，CDA のようにパッチベース型の全結合ニューラルネットでも SRCNN に匹敵する性能が得られることも報告されており，低遅延リアルタイム処理を実現するハードウェア実装を想定したパッチベース型超解像システムのアーキテクチャ設計が必要である．

2.6.2　End-to-End 型学習を活かした性能改善

CDA は，ハードウェア化をすればリアルタイム性を担保できる可能性を持ったアーキテクチャであるが，復元性能という面では SRCNN 9-5-5 に劣っており，更なる改善が望まれる．性能改善の手段としては，深層化によるアプローチがまず考えられるが，コストパフォーマンスという観点からリアルタイム処理では受け入れられない．一方で，補間画素位置を考慮した性能改善については従来の CNN を前提とした検討しか報告されておらず，パッチベース型ニューラルネットに適用するための検討が必要であり，それによる性能改善の期待も高い．また，CDA は二つの Auto-Encoder を形成することを前提としており，シンプルな End-to-End 学習とは異なっているが，より小さいネットワーク規模においても高い性能を得ることが可能であることを示唆している．そこで，超解像という問題を極力単純なネットワーク構造で，End-to-End 学習するという枠組みの中でさらに機械学習による最適化度合いを高める検討には，リアルタイム処理に向けた大きな意義があるといえる．

2.7　結言

本章では，本研究で目指す超解像が持つべき特性である低遅延リアルタイム処理について，その概要を映像表示システムの開発という観点から説明し，その上で，従来の機械学習型１枚超解像について，低遅延リアルタイム処理の実現という観点から，性能と課題について議論していった．本章で述べた内容をまとめると，以下の通りとなる．

2.2 映像表示システムにおける低遅延リアルタイム処理では，ディスプレイのような映像表示機器において，「超解像」等の特殊な映像処理モジュールの最適な挿入位置が汎用スケーラの後段となることを述べた．また，処理対象としては拡大後の画像を対象とすることで，汎用性を高められるメリットがあることを述べた．さらに，表示機器における「低遅延リアルタイム処理」の意味について説明し，特に高解像度映像への対応が必要なディスプレイにおいては ASIC/FPGA で代表される LSI や GPU による並列演算が必要であるということを述べた．

2.3 機械学習による辞書型超解像では，近年，１枚超解像の分野で最も研究な盛んな分野である機械学習型超解像のさきがけとなった W.T. Freeman の事例参照型超解像，及び，C.Y. Yang らの ScSR について，低遅延リアルタイム処理の実現という立場から見解を述べた．具体的には W.T. Freeman らの手法は，幅広い画像で適用するには膨大な事例データベースが必要であり，そのメモリや探索コストが課題であった．一方，C.Y. Yang らの手法はスパース表現によって圧縮された辞書を用いて少ないデータベースから広範囲の自然画像を復元することを可能としたが，一方で，オンラインでスパース表現を実現するには反復演算が必要であり，リアルタイム処理の実現という面で課題があることを述べた．

2.4 畳み込みニューラルネットによる超解像（SRCNN）では，近年のニューラルネット型超解像のさきがけとなった SRCNN について概略を説明し，辞書型超解像をさらに超解像問題に対して最適化させたフィードフォワード型の手法であることを述べた．一方で，低遅延リアルタイム処理という観点から見れば，処理は単純だが遅延時間や演算量については課題があることを述べ，実際に現状のハードウェアコストに関する見積もりを示した．

2.5 SRCNN 以降のニューラルネット型超解像では，ポスト SRCNN の技術動向として，実装コストは増大するがさらなる深層化によって性能向上した VDSR や，処理は複雑化するが補間画素位置を考慮して性能改善した EPCN，そして，性能はそれほど高くないが，パッチベースで処理も軽い CDA を紹介し，それぞれに対する課題や見解を述べた．

2.6 本研究で取り組む課題では，上記関連手法の提案や課題を踏まえ，低遅延リアルタイム処理の実現という観点で，本研究が検討すべき課題とその項目について述べた．具体的には，低遅延リアルタイム処理に適したアーキテクチャ設計と，End-to-End 型学習を活かした性能改善をその項目として示した．

参考文献

[2-1] C. Dong, C.C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” European Conference on Computer Vision, vol.8692, pp.184–199, Springer, 2014.

[2-2] C.-Y. Yang, C. Ma, and M.-H. Yang, “Single-image super-resolution: A benchmark,” European Conference on Computer Vision, vol.8692, pp.372–386, Springer, 2014.

[2-3] W.T. Freeman, T.R. Jones, and E.C. Pasztor, “Example-based super-resolution,” IEEE Computer graphics and Applications, vol.22, no.2, pp.56–65, 2002.

[2-4] 田口安則, 小野利幸, 三田雄志, and 井田孝, “画像超解像のための閉ループ学習による代表事例の学習方法,” 電子情報通信学会論文誌 D, vol.92, no.6, pp.831-842, 2009.

[2-5] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution as sparse representation of raw image patches,” Computer Vision and Pattern Recognition, 2008, CVPR 2008, IEEE Conference on, pp.1–8, IEEE, 2008.

[2-6] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” Signals, Systems and Computers, 1993, 1993 Conference Record of The Twenty-Seventh Asilomar Conference on, pp.40–44, IEEE, 1993.

[2-7] C. Dong, C.C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol.38, no.2, pp.295–307, 2016.

[2-8] S.Y. Kim and P. Bindu, “Realizing real-time deep learning-based super-resolution applications on integrated gpus,” Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on, pp.693–696, IEEE, 2016.

[2-9] NVIDIA Quadro,
https://www.nvidia.com/ja-jp/design-visualization/quadro/

[2-10] Xilinx, Kintex UltraScale,
https://japan.xilinx.com/products/silicon-devices/fpga/kintex-ultrascale.html.

[2-11] Xilinx ホワイトペーパー：浮動小数点と固定少数点, WP491
https://japan.xilinx.com/support/documentation/white_papers/j_wp491-floating-to-fixed-point.pdf

[2-12] J. Kim, J.K. Lee, and K.M. Lee, “Accurate image super-resolution using very deep convolutional networks,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1646–1654, 2016.

[2-13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.770–778, 2016.

[2-14] W. Shi, J. Caballero, F. Husz´ar, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1874–1883, 2016.

[2-15] 大谷，加藤，”4 並列の畳み込みニューラルネットワークを用いた超解像,” 電子情報通信学会論文誌 2019/5 Vol. J99-D No.5.
[2-16] K. Zeng, J. Yu, R. Wang, C. Li, and D. Tao, “Coupled deep autoencoder for single image super-resolution,” IEEE Trans. Cybern., vol.47, no.1, pp.27–37, 2017.

第 3 章 パッチベース型 DNN による超解像システム

3.1 緒言

前章では，近年の機械学習型超解像分野における技術革新を振り返りながら低遅延リアルタイム処理を実現する上での課題をそれぞれ述べた．また，2.6 節では従来技術が持つ課題を受けて，本研究で取り組むべき課題を集約して示した．本章では，これら課題の解決に向けた提案手法を提示する．具体的には，パッチベース型のニューラルネットによって構成された低遅延リアルタイム処理が可能な超解像システムを提案する．また，性能改善のために不要な制約や自由度を極力取り除いたシンプルな End-to-End 学習システムも提案する．

3.2 PDNN（Patch-based Deep Neural Network）の設計

2.4.3 項でも述べた通り，CNN をベースとしたニューラルネットは処理の中間結果が画像サイズそのものとなるため，実装コストや遅延時間に課題があった．提案手法では，超解像の低遅延リアルタイム処理をパッチベース型のニューラルネット（PDNN: Patch-based Deep Neural Network）により実現する．本節では，提案する PDNN の具体的な設計方法について述べる．

3.2.1 入出力定義

ニューラルネットを設計する上で，まず始めに考えなければならないのは，対象のニューラルネットが何を推定するものであるのかという点である．これはニューラルネットを含むシステム全体の中でのニューラルネットの役割を定めることで決まり，それに基づき

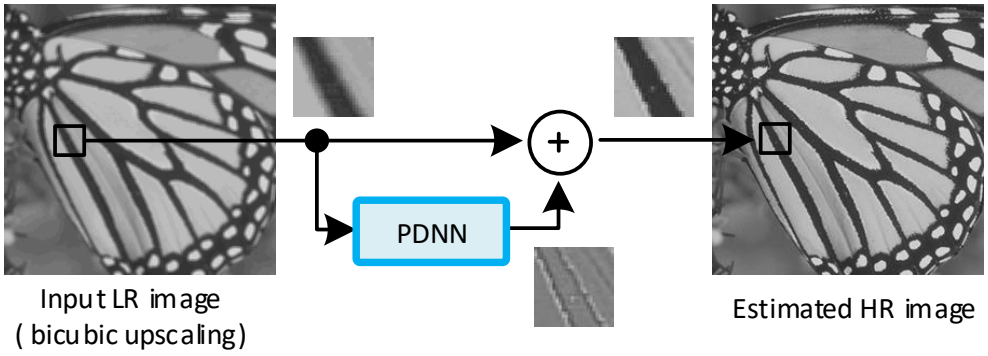


図 3.2.1：PDNN の入出力定義

ニューラルネットの入出力も定義される．例えば，2.5 節で述べた SRCNN [3-1] では，ニューラルネットの入力は低解像度品質の画像であり，出力である学習対象は高解像度品質の画像そのものであった．一方で，2.5.1 項で述べた VDSR [3-2] では ResNet [3-3] 構造を取り入れ，高解像度品質画像と低解像度品質画像の残差成分を学習対象としていた．提案手法においても ResNet 構造を取り入れた 図 3.2.1 で示すシステム構成によって PDNN の役割を定める．

いま，ある観測された低品質画像パッチを $y \in \mathbb{R}^n$，教師データに該当する y に対応した高品質画像パッチを $x \in \mathbb{R}^n$，そして，y から x を推定した結果を $\hat{x} \in \mathbb{R}^n$ とすると，学習時，及び，推定時における PDNN の役割を式（3.2.1），式（3.2.2）で定義する．

$$at\ learning : PDNN(y) \leftarrow x - y \qquad (3.2.1)$$

$$at\ estimation : \hat{x} = PDNN(y) + y \qquad (3.2.2)$$

PDNN の学習手順に関する詳しい内容は 3.3 節で述べるが，このような Residual Net（ResNet）構成をとることによって，PDNN の出力は高解像度品質画像と低解像度品質画像の残差成分（Residual）の推定値となり，低解像度化によって失われた成分そのものに推定リソースを集中させることができる．図 3.2.2 はテスト画像 lena の場合を例に学習対象となる残差成分を図示したものである．ここでの高品質画像と低品質画像の残差成分とは，拡大前の低品質画像では表現できなかった高周波成分に該当し，PDNN としてはこの残差成分の推定を目標にすることを意味する．こうすることで，たとえ PDNN の出力がゼロであったとしても入力である低周波成分 y の品質は容易にクリアできる．また，将来的にディスプレイの搭載機能として超解像を実装する場合，効果の強弱設定は実用上必要になってくることが多いため，原画像成分と付加情報成分を予め分けることができれば，付加情報にゲインを掛け合わせることで処理前から処理後への変換度合いをシームレスに可変させられるようになるという利点も得られる．

図 3.2.2： 提案法における PDNN の学習対象

3.2.2　出力層（復元部）

入出力定義の次に設計したいのは出力である復元部である．前項で述べたとおり，PDNN によって生成される信号は高周波成分であり，バイアス成分は不要となるはずである．従って，式（3.2.3）によって，PDNN の出力層を定義することが出来る．

$$PDNN(y) = \sum_{i=1}^{f_h} atom_i \times p(atom_i|y) \qquad (3.2.3)$$

ここで，$atom \in \mathbb{R}^n$ は入出力である x，y と同次元数の列ベクトルであり，推定目標となる高周波成分を構築するための最小構成要素，すなわち，基底ベクトルといえる．また，$p(atom_i|y)$ は入力信号 y が与えられた時における出力値において基底ベクトル $atom_i$ の寄与を示す重み係数である．

次に，式（3.2.3）で定義された出力をニューラルネットの出力として扱っていくための形式変換を考える．図 3.2.3 は式（3.2.3）を図示したものであるが，出力を構成する成分は予め設定された基底ベクトル（atom）群と，それら基底ベクトル群に掛け合わされる

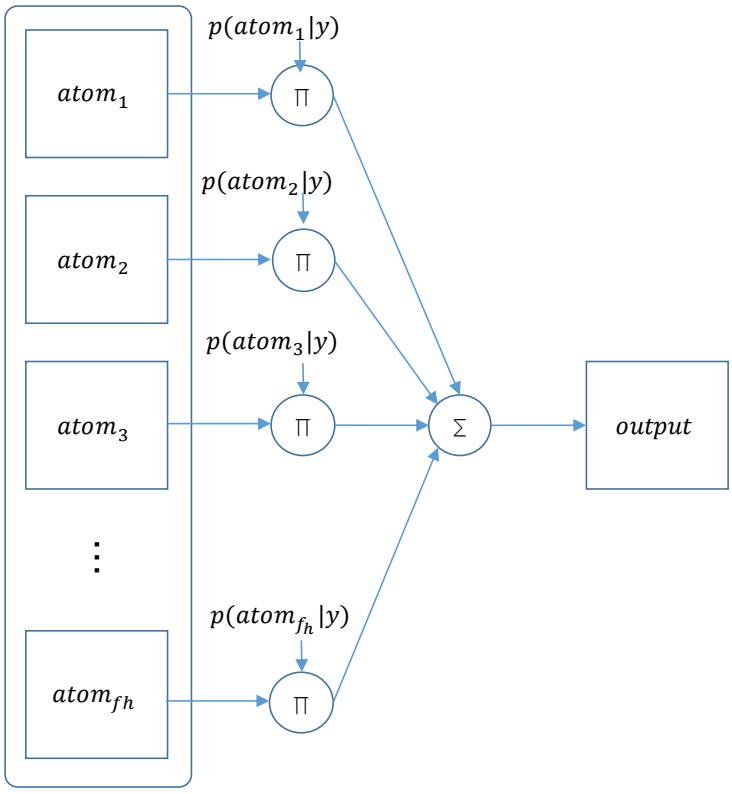


図 3.2.3：復元部の構成

重み係数 $p(atom_i|y)$ の集合によって表される．また，出力は当然ながら入力に依存するが，入力に依存する部分，すなわち，ニューラルネット上を流れる信号として表現されるべき対象は重み係数 $p(atom_i|y)$ であり，一方で，予め用意されるべき基底ベクトルはニューラルネット上における重み行列として表現されるべきであるという指針が定まる．

上記指針に基づき，まずは，全ての基底ベクトルの集合として，$n \times f_h$ 次元の重み行列 W_{atom} を式（3.2.4）で定義する．

$$W_{\text{atom}} = [\, atom_1 \; atom_2 \ldots atom_{f_h}]$$ （3.2.4）

さらに，入力に依存した信号成分，すなわち，出力層よりも前段の出力信号となるべき重み係数についても，$f_h \times 1$ 次元の行列 $p(W_{atom}|y)$ として式（3.2.5）で定義する．

$$p(W_{atom}|y) = \left[\, p(atom_1|y)\, p(atom_2|y) \;\ldots\; p\left(atom_{f_h}|y\right) \right]^T$$ （3.2.5）

式（3.2.4）及び，式（3.2.5）を式（3.2.3）に代入することで，PDNN の出力値は次式によって表すことができる．

$$PDNN(y) = W_{atom} \times p(W_{atom}|y)$$ （3.2.6）

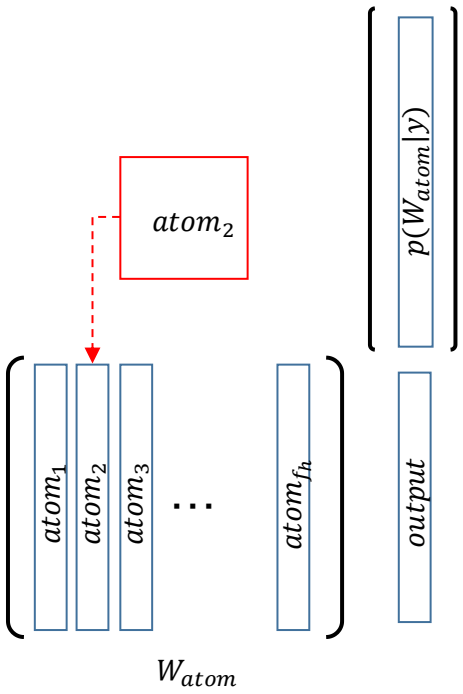


図 3.2.4： 変換された復元層

ただし，ここでの ”×” とは単純な行列の積を表しており，$PDNN(y)$ は $n \times 1$ 次元の行列として出力される．

図 3.2.4 は式（3.2.6）を図示したものであり，図 3.2.3 と等価な処理を示している．つまり，列ベクトルに変換された基底ベクトル群を重み行列 W_{atom} とし，そこに各基底ベクトルの係数が格納された列ベクトルである前段の出力値 $p(W_{atom}|y)$ を行列の積として掛け合わされることで出力のパッチ画像と同次元の列ベクトルが生成される．このような形式変換をすることで，出力層をニューラルネットに適した形式に変換することができる．つまり，W_{atom} は PDNN 最終段における学習対象となる重み行列の一つであり，$p(W_{atom}|y)$ は出力層前段のネットワーク出力値と見做すことができる．

3.2.3 入力層（特徴抽出部）

出力層の次に設計したいのは出力層と対称的な構成で表現される入力層である．PDNN における入力層の役割としては，前項における $p(W_{atom}|y)$ を推定するために必要となる情報を抽出することにある．また，一般に超解像で復元したい情報は低解像度品質画像では表現できなかった高周波成分であることから，復元したい情報の足がかりとなる情報もまた入力画像の高周波成分に多く含まれることが予想される．以上の理由から，PDNN の入力層としては 図 3.2.5 で示す特徴抽出機構を設計する．本特徴抽出部では，入力画像パッチに対し複数の特徴抽出フィルターを用意し，各フィルターと画像パッチの内積計算で

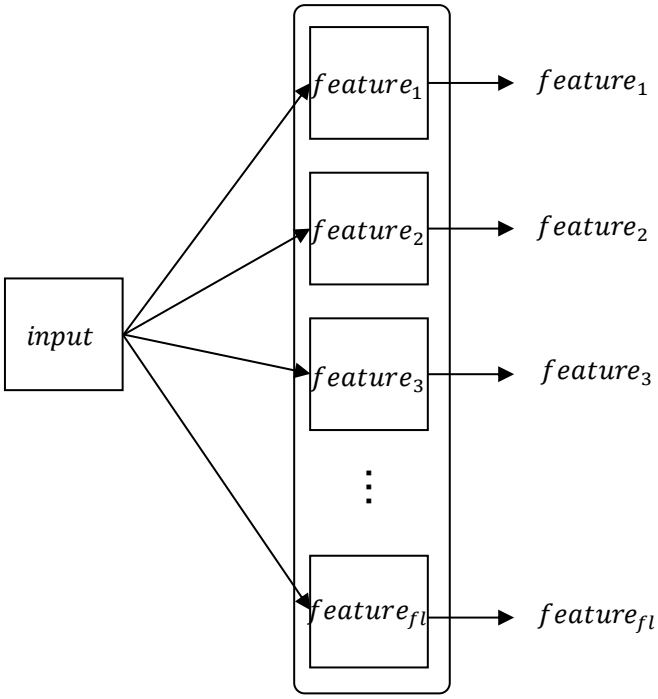


図 3.2.5： 特徴抽出部の構成

定まる出力を特徴量として出力する．このとき，特徴抽出部は式（3.2.7）で表される．

$$\bigcup_{j=1}^{f_l} filter_j \times y \qquad (3.2.7)$$

ただし，$filter_j \in \mathbb{R}^n$ は j 番目の特徴フィルターを意味し，入出力である x，y と同次元数の列ベクトルである．また，f_l は特徴抽出フィルターの総数である．

ここで，前項同様，式（3.2.7）をニューラルネットで表現することを考える．まず，入力信号 y に依存するものと依存しないものに分けることを考えれば，特徴量は当然ながら入力信号 y に応じて変化すべきである．一方で，特徴量フィルターの係数値は予め学習済みのものを用意しておいても良い．以上の方針より，各フィルター係数を束ねた行列 $f_l \times n$ 次元の重み行列 W_{filter} を式（3.2.8）で定義する．

$$W_{\text{filter}} = \left[filter_1\ filter_2 \ldots\ filter_{f_l} \right] \qquad (3.2.8)$$

式（3.2.7）と式（3.2.8）から，特徴抽出部を式（3.2.9）で定義する．

$$Features\,(\mathrm{y}) = soft\ sign\,(W_{filter} \times y) \qquad (3.2.9)$$

ただし，softsign は次式で定義される活性化関数である．

$$softsign\,(\alpha) = \frac{\alpha}{1 + |\alpha|} \qquad (3.2.10)$$

図 3.2.6 は式（3.2.9）におけるW_{filter} の役割を図示したものであるが，出力層と同様に，PDNN 入力層においても，フィルターを列ベクトルとする重み行列を構成することで，特徴抽出部における入力を y，出力を特徴量 $Features$ と見做したネットワーク表現が可能となる．なお，活性化関数に softsign を選択した理由としては，抽出したい情報が高周波成分であることから，伝達信号も正負の値を許容し，かつ，ゼロ付近に最大勾配を持った活性化関数が相応しいという判断である．また，非線形の活性化関数が加わることによって，単なる線形写像では表現できなかった非線形写像を可能とし，より高次な推定を可能にする意図もある．

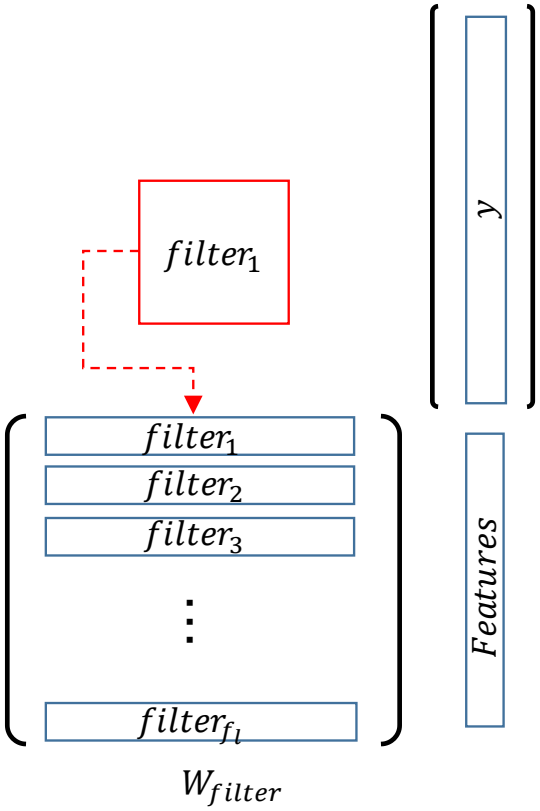


図 3.2.6：変換された特徴抽出部

3.2.4　中間層（推定部）

入力と出力が定まれば，最後に設計すべきは中間層に相当する推定部である．中間層の出力として推定すべき値は式（3.2.5）で定義された $p(W_{atom}|y)$ であり，そのための材料として与えられるのは入力層の出力である式（3.2.9）で定義された $Features$ である．いまコストパフォーマンスを重視して，中間層を 1 層からなる全結合型ニューラルネットで構成することを考えれば，中間層の定義は式（3.2.11）となる．

$$p(W_{atom}|y) = soft\ sign\left(W_{convert} \times Features(y)\right) \qquad (3.2.11)$$

ただし，W_{convert} は $f_h \times f_l$ 次元の重み行列であり，f_l 次元の特徴量を f_h 次元の係数行列に変換する役割を担っている．また，softsign についても入力層と同様の理由から用いる．なお，図 3.2.7 は式（3.2.11）における行列のサイズ関係を図示したものである．

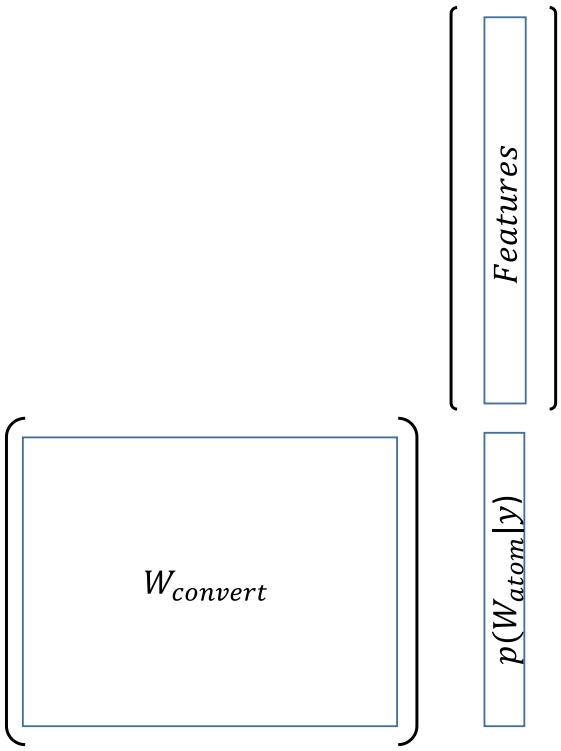


図 3.2.7：推定部の構成

3.2.5　PDNN の内部構成

本項では，本節で述べた PDNN の内部構成をまとめて記述する．PDNN は図 3.2.8 で示したとおり，特徴量の抽出を行う第 1 層と，特徴量の変換を行う第 2 層，出力画像を合成する第 3 層によって構成され，それぞれ以下の通り記述される．

第 1 層（特徴抽出部）：

$$Features\,(\mathrm{y}) = soft\ sign\,(\,W_{filter} \times y\,) \qquad (\,3.2.12\,)$$

第 2 層（推定部）：

$$p(W_{atom}|y) = soft\ sign\,(\,W_{convert} \times Features(y)\,) \qquad (\,3.2.13\,)$$

第 3 層（復元部）：

$$PDNN(y) = W_{atom} \times p(W_{atom}|y) \qquad (\,3.2.14\,)$$

また，PDNN の入出力特性は重み行列である{ W_{filter} , W_{convert} , W_{atom} } によって決定される．これらの重み行列値の決定方法については 3.3 節で詳しく述べる．

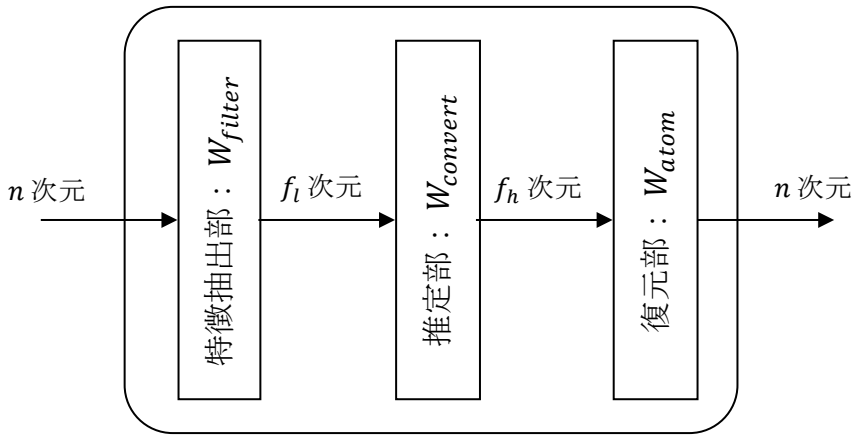


図 3.2.8：PDNN の内部構成

3.3　学習による重み行列の最適化

3.2 節では PDNN の内部構成，及び，アーキテクチャ設計について述べた．3.2 節の結論として PDNN には W_{filter}, W_{convert}, W_{atom} で表される 3 つの重み行列が含まれており，これらの重み行列は学習によって最適化される必要があることを述べた．学習は最終的なシステム性能を決める重要な要因となる．本節では，これら重み行列の学習方法について述べる．

3.3.1　補間画素位置に基づく学習対象の限定化

一般に，ニューラルネットを学習させる上で，問題を如何に単純化できるかは最終的に期待できるパフォーマンスに大きく影響する．なぜならば，ニューラルネットの学習は損失関数と呼ばれる評価関数を最小（あるいは最大）にすべく行われるものであるが，基本的にはそれらの評価関数の値は学習データセットと呼ばれる様々なパッチから導き出される正解データと推定データの平均的な差を小さくしていく方向に働くからである．つまり，正解データそのものが持つばらつき以上の精度を推定結果として期待することは原理的に難しく，その意味で，正解データのばらつきを如何に抑えるか，すなわち，問題を如何に単純化できるかという視点はニューラルネットの推定精度向上を検討する上で重要となる．

このことは超解像分野の学習をする際にも同様に重要と捉えるべきである．既に 2.5.2 項でも触れた通り，拡大画像の補間画素位置を考慮することによって正解データの ”冗長な” ばらつきを抑制するという試みが成されており，その重要性は示されていた．本提案手法においてもその思想を継承し，パッチベース処理における学習対象のばらつき抑制を

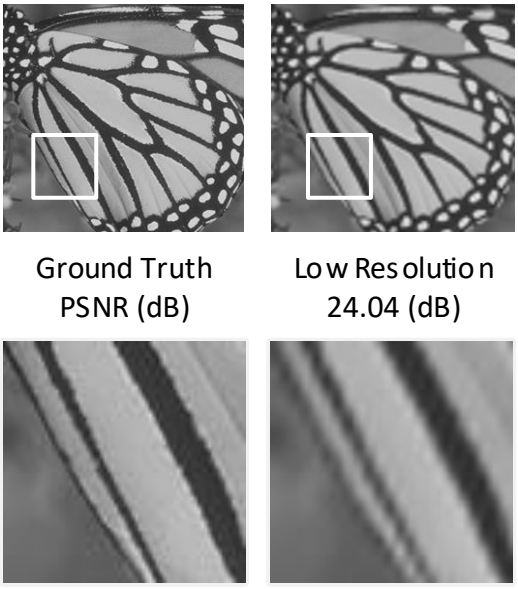


図 3.3.1： 低解像度画像の品質

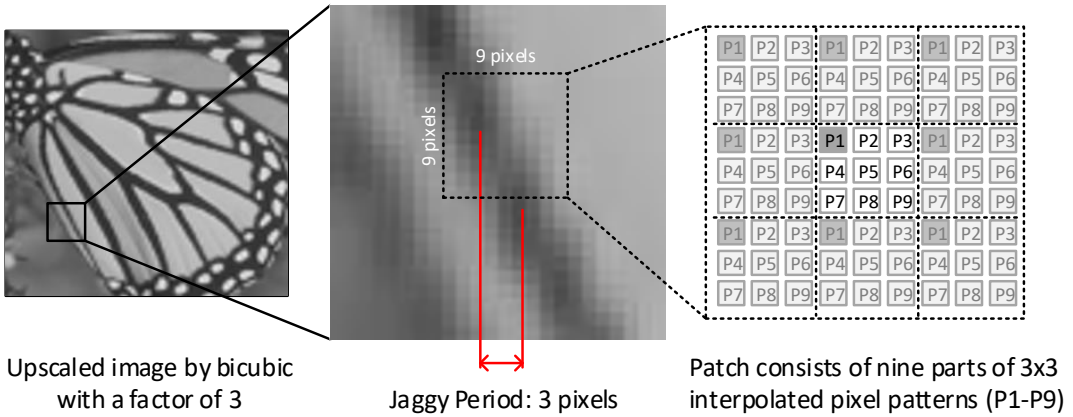


図 3.3.2： 拡大画像に含まれるジャギー

試みた．

図 3.3.1 は 3 倍拡大を例にした高解像度品質画像（ Ground Truth ）と低解像度品質画像（Low Resolution）の画質を比較したものである．これらの画質を比較すると，低解像度品質画像には高解像度品質画像に含まれていた高周波成分が失われていることが分かると共に，高解像度品質画像には無かったジャギーと呼ばれるアーチファクト（人工的に発生する模様のことで，ここでは周期的にみられる斜め線のことを指す）が現れていることが確認される．低解像度品質画像はオリジナル画像を 1/3 倍のサイズにバイキュービック縮小した後，3 倍のサイズにバイキュービック拡大で戻すといった処理で生成されており，ジャギーの発生は拡大時におけるバイキュービック補間において発生することが分かっている．実際，図 3.3.2 の中央画像は，左側に示した低解像度品質画像（拡大率： 3）の一部を拡大表示したものであるが，バイキュービック拡大の比率である 3 pixel 周期でのジャギーが視認できる．また，図 3.3.2 の右側に示された図は中央の 9 × 9 pixel のパッチ領域に対応したバイキュービック画像の補間式を模式的に表したものであるが，3 倍拡大の場合，P1 ~ P9 として表される 9 パターンの混合式によって形成されることになる．また，いずれの補間パターンも縦横 3 pixel の間隔で繰り返されることになる．

ここで上記パッチ領域を PDNN の入力と捉えた場合を考える．PDNN の出力として推定すべきは当然ながら同一箇所におけるオリジナル画像のパッチ領域となる．また，学習を行う上では出来るだけ多くの想定パターンを含める必要があるため，一般に 1 枚の画像から多くの画像パッチのサンプルを抽出する．このとき，単純で最も多くのサンプルを入手する方法としては，上記のパッチ抽出領域を上下，或いは左右に 1 pixel 単位でずらしながら新たなサンプルデータを入手する方法となる，しかし，その方法では各ニューロンの位置（すなわち，パッチ画像の相対座標位置）とバイキュービック補間式の関係性は得られなくなる．つまり，拡大済みの低品質画像には図 3.3.2 の右側に示されたような 9 種類の補間パターンが規則正しく周期的に並んでいるという先見情報を有効に活用しないまま学習されることを意味し，正解データに冗長なばらつきが含まれてしまうことになる．

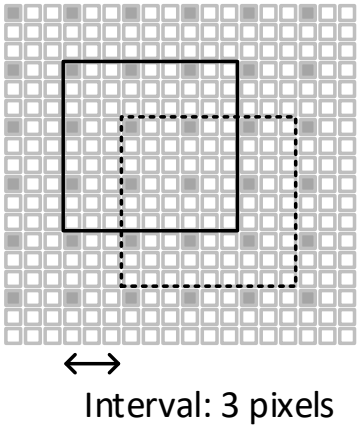


図 3.3.3：補間画素位置を考慮した画像パッチの切り出し

上記問題を解決するには，画像パッチを抽出する際の切り出し位置を拡大率に応じたステップ幅に設定すればよい．例えば，図 3.3.3 は 図 3.3.2 の右側の補間パターンをより広い視点で表現したものである．灰色で塗りつぶされたピクセルは 図 3.3.2 の P1 に相当するが，実線で示されたパッチ領域と点線で示されたパッチ領域内に含まれる補間式のパターンは対応が取れており，いずれもパッチ画像の左上から右に向かって順に P1, P2, P3, P1, P2, P3, P1, P2, P3 と特定の補間式に対応したパターンが並ぶことになる．つまり，拡大率と同じ間隔で切り出されたパッチ画像はいずれもパッチ画像内における相対座標とバイキュービック補間式における補間パターンの対応が一致し，PDNN の各ニューロンには特定の補間画素パターンに限定された情報が入力されることになる．

上記の通り，提案手法では本位置合わせを行うことで学習時におけるばらつきを軽減している．このような学習条件の単純化は限られたリソースで推定精度向上をしていく上で重要である．なお，本位置合わせの性能に対する効果については 4.3.1 項で詳しく述べる．

3.3.2 パッチベース手法における位置あわせの利点

3.3.1 項で述べた通り，パッチベース型の PDNN では，学習対象であるパッチ画像の切り出し位置を拡大率に応じて変化させることで，各ニューロンとの対応関係を限定することが出来る．このようなパッチベース手法における位置あわせの実現は非常に簡易であり，しかも，ニューラルネットの規模を拡大する必要なく学習精度を向上させられるのはパッチベース手法の利点であるといえる．なぜならば，2.5.2 節で述べた CNN をベースとした手法［3-4, 3-5］では，補間画素との対応関係をとるためにはネットワークそのものを増加させる必要があったからである．これは，CNN がフィルターベース処理であることに起因している．

また，本位置あわせ手法のもう一つのメリットとして，復元時の処理も簡略できるという点が挙げられる．これは学習時においてパッチ画像の各位置と各ニューロンのパターンを一致させるということは，復元時においても必然的に同様の状態を再現する必要があることから生まれる制約といえるが，結果として復元時も拡大率のステップでパッチを飛ばしながら切り出して復元処理をすることになる．これは 1 pixel 毎にシフトしながら処理をするよりも処理を拡大率の 2 乗オーダーで削減することができることを意味する．なお，任意画像における画像復元の詳しい原理と手順については 3.4 節で詳しく述べる．

3.3.3 損失関数

3.3.1 項で述べた通り，PDNN の学習においては学習対象であるパッチ画像のサンプル抽出位置に気をつける必要はあるものの，損失関数の設定という観点では，式（3.2.2）で定義した PDNN の役割をシンプルに表現する損失関数を設定すれば良い．具体的には式（3.3.1）で定義された学習対象パラメータ Θ に対し，式（3.3.2）で表現される損失関数を定義する．

$$\Theta = \{W_{filter}, W_{convert}, W_{atom}\} \tag{3.3.1}$$

$$loss\,(\Theta) = \frac{1}{N_s}\sum_{n=1}^{N_s} \|PDNN(y_n;\Theta) - (x_n - y_n)\|^2 \tag{3.3.2}$$

ここで，$PDNN(y_n;\Theta)$ は，学習パラメータ Θ が設定され，かつ，y_n が入力されたときにおける PDNN の出力値を意味する．また，N_s は学習データセットに含まれるサンプルの総数を意味する．式（3.3.2）の損失関数を設定することで，3.2.1 項で述べたとおり，高解像度品質画像と低解像度品質画像の残差成分，すなわち，低解像度化によって失われた高周波成分が学習対象となる．

また，上記損失関数は 式（3.3.3）で一般的に表される画像 I, K 間の平均 2 乗誤差（Mean Squared Error：MSE）と同義である．

$$\mathrm{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{3.3.3}$$

従って，式（3.3.4）で一般に定義される PSNR を最大にする条件，すなわち，MSE を最小にする条件に合致した損失関数であると捉えられる．PSNR は画像の類似性を表す最も一般的な客観評価指標であり，第 4 章の性能比較でも復元性能を比較する手段として，本指標を利用する．

$$PSNR = 10\log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{3.3.4}$$

$$= 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

ここで, MAX_I は画像 I の対象ビット深度における最大値 (8bit の場合 255) であり, PSNR の単位は dB である.

3.3.4 学習手順

PDNN を学習する上で必要な事項については既に述べた．本節では，PDNN 学習の具体的なステップについて述べる.

1. 学習用画像の生成

まず, 教師画像である高解像度品質（High Resolution quality: HR）画像 を用意する. 対象は超解像処理を行いたいものに最も近いものが良いが，学術的な比較を行う際は学習データとしては同じものを選択するのが最もフェアである．例えば，SRCNN［3-1］で使用されている 91 images 等である．HR 画像セットが定まったら，低解像度品質（ Low Resolution quality: LR）画像を生成する．これは，HR 画像を一旦，対象の拡大率でバイキュービック縮小した後，再度，バイキュービック拡大することで実現する．こうすることで，HR 画像とペアになる LR 画像のセットが入手できる.

2. 学習用パッチ画像の生成

次に，ペアとなる HR 画像と LR 画像から，同一箇所を切り抜きしてパッチ画像を生成する．ただし，最適な画像のパッチサイズは拡大率に応じて変化する．理由は 3.4 節で述べるが，拡大率が 2 の場合は 6 ×6 pixel，拡大率が 3 の場合は 9×9 pixel，拡大率が 4 の場合は 12 ×12 pixel 程度が良い．切り出すパッチサイズが決まれば，水平・垂直方向にラスタースキャンしながら HR, LR 画像からペアとなる画像パッチを抽出する．ただし，スキャン時のステップ幅（粒度）は，3.3.1 項で述べた通り，対象の拡大率に応じて調整する．本作業を全ての学習用画像に施すことで，パッチ画像になった学習用データセットが完成する.

3. 学習の実行

最後に, Step.2 で得られた学習用パッチを用いて 式（3.3.2）を最小にする学習を行う. このとき学習対象である重み行列 $\Theta = \{W_{filter}, W_{convert}, W_{atom}\}$ は予め全てランダムな値で初期化を行う．実験条件については第 4 章で述べるが，目安としては平均 0，分散 0.001 で初期化する．また，学習自体は原理的には誤差逆伝播法と呼ばれる最急降下法による反復処理で最適化するが，一般に学習セットは膨大であり 91 images を 6×6 pixel で切り

出した場合でも 500 万組程度の数になる．そのため，コンピューターによる最適化処理をスムーズに行うため，画像セット全体から学習作業を実施する PC の GPU メモリに収まる範囲でランダムに選択されたパッチ画像セットを用いて，繰り返し学習を行う手法であるミニバッチ学習を行う．この際，式（3.3.2）の N_s はバッチ学習時における画像の組数として取り扱う．この時，ランダムに選択されたミニバッチ学習を繰り返し行うことで，擬似的に全学習データセットに対する最適化を行ったものとして見做すわけであるが，当然ながら，選択されたバッチの特性が学習画像セット全体の特性を表しているとは限らず，学習手法としては様々な方法が提案されている．本研究では比較的一般に使用されることが多い Adam Optimizer [3-6] と呼ばれる統計確率的最適化手法を用いて学習を行った．

3.4　PDNN を用いた超解像システム（SR-PDNN）

前節まで述べてきた PDNN はあくまでパッチ画像を入出力としたアーキテクチャであり，任意サイズである実際の画像に超解像処理を施すには PDNN を取り込んだシステムを構築する必要がある．本節では学習済みの PDNN を用いた超解像システム（ Super Resolution via PDNN: SR-PDNN ）について述べる．

3.4.1　システム構成

初めに，図 3.4.1 に示した SR-PDNN のシステム構成図を基に，基本的な流れを説明する．まず，入力である LR 画像から画像パッチ $y \in \mathbb{R}^n$ を抽出し，PDNN へ入力する．PDNN 内部では，第 1 層の特徴抽出部，第 2 層の推定部，第 3 層の復元部を経て，高周波成分に相当する$PDNN(y)$ を出力する．それに低周波成分である入力 y を加算し，推定結果 x として出力画像のパッチ画像を得る．同様にして，画像の左上から順にラスタースキャンしていくことで出力画像全体の推定結果を得ることになる．以降では，本処理についてより詳しく述べる．

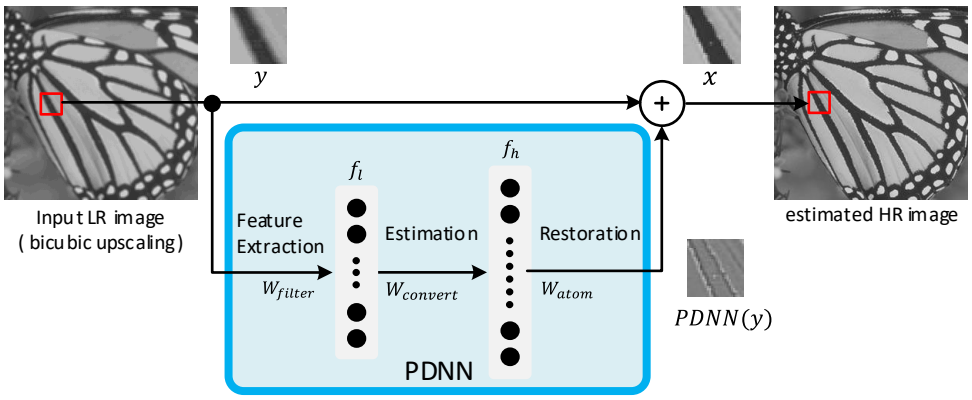


図 3.4.1：SR-PDNN の構成

3.4.2　入出力定義

まず，PDNN の入力を実現するために必要となるパッチ画像の抽出を行うための数学的表現を整理する．いま，LR 画像を現す記号として，列ベクトル $Y \in \mathbb{R}^{N_p}$ を式（3.4.1）で定義する．ただし，N_p は 2 次元画像 *input LR image* のピクセル数であり，画像の幅と高さを掛け合わせた大きさである．

$$Y = \bigcup_{i=1}^{height} \bigcup_{j=1}^{width} input\ LR\ image\ (i,j)\ \in \mathbb{R}^{N_p} \qquad (3.4.1)$$

同様に，超解像処理によって推定される 2 次元画像 estimated $HR\ image$ と対応する列ベクトル $\hat{\mathrm{X}}$ との関係を式（3.4.2）で定義する．

$$\hat{\mathrm{X}} = \bigcup_{i=1}^{height} \bigcup_{j=1}^{width} estimated\ HR\ image\ (i,j)\ \in \mathbb{R}^{N_p} \qquad (3.4.2)$$

通常，画像は上式右辺のように 2 次元配列として表現されることが多いが，ここでは 1 つの画像を列ベクトルとして扱う．画像を列ベクトルにすれば，複数の画像も 2 次元行列で表現できるようになり，3.3.4 項で述べたミニバッチ学習も行いやすくなるからである．また，上記画像の 1 次元ベクトルへの変換は形式的にデータ配列の構造を変更しているのみであり，画像情報としては 1 対 1 に対応している．これはパッチ画像を列ベクトルとして扱う場合と同様である．

3.4.3 パッチ画像の抽出

列ベクトルで表現された画像 Y から p 番目の画素を中心としたパッチ領域を抽出するオペレータ R_p を次式で定義する．

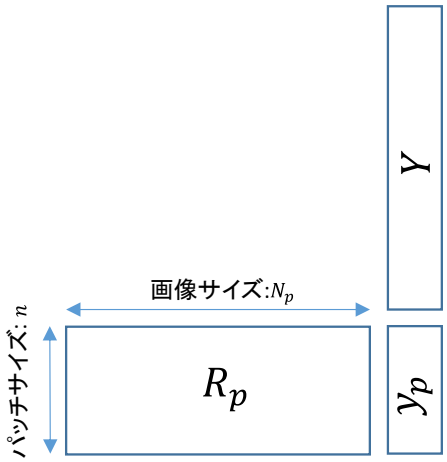


図 3.4.2：パッチ抽出オペレータ R_p

$$y_p = R_p Y \qquad (3.4.3)$$

ただし，$p = 1, 2, \ldots, N_p$ である．図 3.4.2 は $n \times N_p$ 次元の行列 R_p の役割とサイズを図示したものである．図の通り，R_p の行数はパッチサイズに対応しており，列数は画像サイズ数に対応している．従って，各行毎にただ 1 つだけ，抽出対象ピクセル位置に対応した箇所の値が 1 となり，それ以外は 0 となる．このような表現を用いることで，任意箇所でのパッチ画像の抽出を表現することが可能となるが，ここでは対象画素 p を中心としたパッチ領域の抽出を想定している．実際には 3.3.4 項と同様に，拡大率が 2 の場合は 6×6 pixel，拡大率が 3 の場合は 9×9 pixel，拡大率が 4 の場合は 12×12 pixel をパッチサイズとする．

3.4.4　パッチ画像の復元と合成

パッチ抽出オペレータ R_p によって，LR 画像のパッチ画像である y_p が抽出できれば，対象位置における推定 HR 画像パッチ $\hat{x}_p$ は次式で表される．

$$\hat{x}_p = PDNN\left(y_p\right) + y_p \qquad (3.4.4)$$

次に，推定 HR 画像パッチ $\hat{x}_p$ を元の画像サイズの空間に写像する．これは，図 3.4.3 でも示したとおり，パッチ抽出オペレータ R_p の転置によって可能である．つまり，入力画像 Y と同一サイズの画像列ベクトルに対して，y_p が抽出された位置と同じ位置に $\hat{x}_p$ を貼り付ける操作は式（3.4.5）によって実現できる．

$$\hat{X}_p = R_p^T \hat{x}_p \qquad (3.4.5)$$

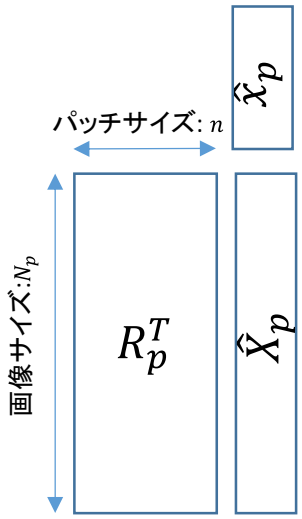


図 3.4.3：パッチ復元オペレータ R_p^T

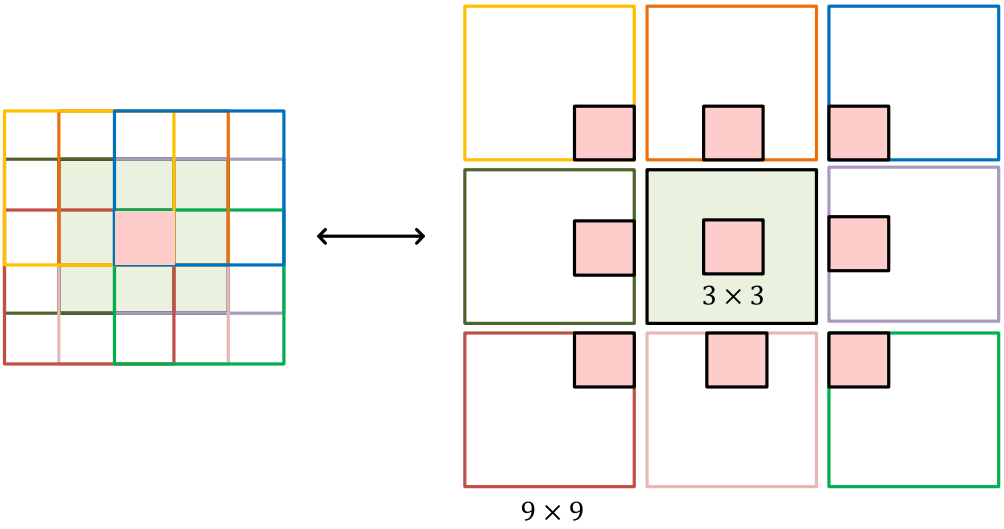


図 3.4.4：パッチ合成時のオーバーラップ

ただし，上式によって実現される $\hat{X}_p$ は対象パッチエリア以外の値は全て 0 が埋まった状態となる．

また，推定 HR 画像は，全ての画素位置における画像パッチの復元結果の重ね合わせで表現できることから，式（3.4.6）で表される合成式によって生成される．

$$\begin{aligned}\hat{\mathrm{X}} &= \left(\sum_p R_p^T R_p\right)^{-1} \sum_p \hat{X}_p \\ &= \left(\sum_p R_p^T R_p\right)^{-1} \sum_p R_p^T \hat{\mathrm{x}}_\mathrm{p}\end{aligned} \qquad (3.4.6)$$

ただし，$\left(\sum_p R_p^T R_p\right)^{-1}$ は，パッチ合成時に生じるオーバーラップ部分の平均化を意味する．例として 図 3.4.4 は拡大率が 3 でパッチサイズが 9×9 pixel の場合を図示したものであるが，左側の画像空間にあるピンク色で示した 3×3 pixel のエリアは右側の 9 種類で表されるパッチ画像の一部として抽出される．従って，合成時にはそれぞれの復元結果を重ね合わせた後，平均化（9 で割る）する必要がある．このようにパッチサイズを拡大率の 3 倍に設定すると，拡大率が 2 の場合も 4 の場合も同様に周囲 8 近傍に “拡大率 × 拡大率” サイズの小領域（図 3.4.3 でのピンク領域）が含まれる構図になる．つまり，画像パッチは 9 つの小領域によって構成されることになり，かつ，それら全ての領域の結果を重ね合わせた結果が最終的な推定結果となっている．つまり，確率的な平均化作用が得ら

れることを意味しており，より多くのサンプルによって誤差が平均化されるという点で，学習による損失関数と相性が良い．また，誤差の平均化作用はパッチサイズを拡げることで効果の向上が期待できるが，一方でパッチサイズの拡大は超解像としての生成パターンの増加を意味していることから難易度の向上やネットワーク規模の増大へと繋がる．従って，拡大率の 3 倍にパッチサイズを設定することはネットワーク規模を小さくしながらも高い平均化作用による性能向上を狙える最適なサイズといえる．これが 3.3.4 項で示した拡大率に対するパッチ画像サイズ設定の根拠となる．

3.4.5　ハードウェア実装コストの見積もり

本項では SR-PDNN のハードウェア実装コストの算出過程について述べる．まず，前提としてハードウェア実装のコストを見積もる際には 2.2 節で述べたとおり，画像情報がラスタースキャンされたストリーミングとして伝達されてくるため，実装回路としてはスループットさえ確保できれば最小単位（例えば 1pixel のみを実現するなど）で実装すればよい．つまり，PDNN のそのものの実装コストを見積もることがハードウェア実装コストを見積もることに相当する．

図 3.4.5 は一例として，拡大率が 2，パッチサイズが 6×6 pixel，特徴量の次元 $f_l = 200$ ，復元部における基底ベクトルの数 $f_h = 800$ の場合における PDNN の実装コストを想定している．まず第 1 層のメモリコストとしては，入力のパッチサイズに相当することから，以下のメモリが必要になる．

第 1 層のメモリコスト：　　$6 \times 6 = 36$ pixel

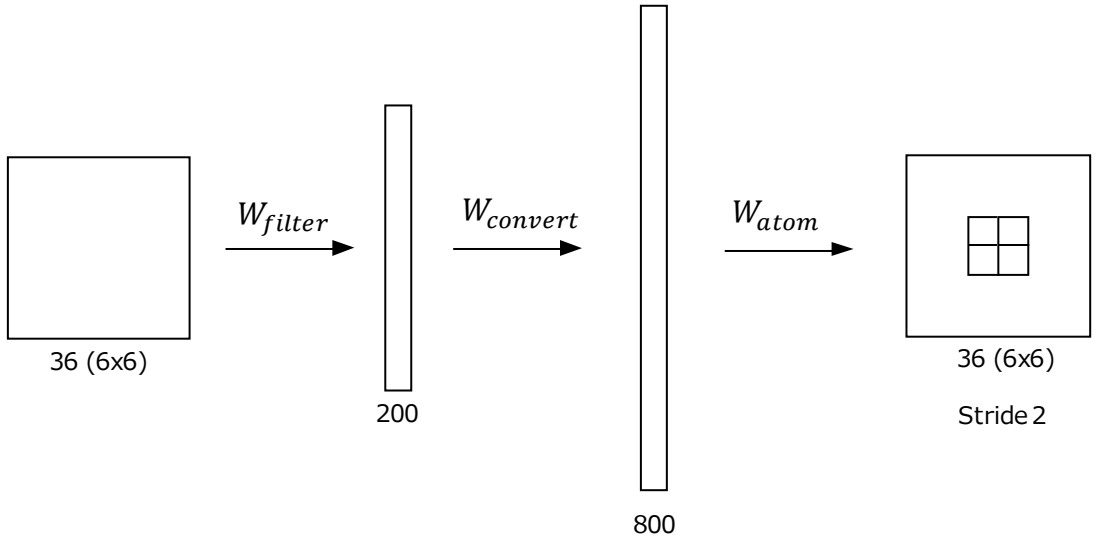


図 3.4.5：SR-PDNN のハードウェア実装コスト

なお，ここでは便宜上 pixel という単位を用いているが，実際はニューラルネット上における 1 入力もしくは 1 出力に対応した値であり，一般には実数値をとることから 32bit 程度の情報量を想定するのが妥当である．また，積和演算コストについては，W_{filter} と入力 y の行列積である事から，W_{filter} の行列サイズと等しくなり，以下となる．

第 1 層の積和演算コスト：　　$36 \times 200 = 7{,}200$ step

なお，ここでの step とは乗算回数を示しており，実際にはハードウェア上において乗算ロジックが置かれる個数に対応している．次に，第 2 層のメモリコスト，すなわち，第 1 層の出力値は $f_l = 200$ と等しいため，以下となる．

第 2 層のメモリコスト：　　200 pixel

また，第 2 層の積和演算コストは，$W_{convert}$ の行列サイズと等しくなるため，以下となる．

第 2 層の積和演算コスト：　　$200 \times 800 = 160{,}000$ step

最後に，第 3 層のメモリコスト，積和演算コストも同様に算出すると，以下となる．

第 3 層のメモリコスト：　　800 pixel
第 3 層の積和演算コスト：　　$800 \times 36 = 28{,}800$ step

これらの合計は，メモリコストとしては 1036 pixel，積和演算コストとしては 196,000 step となる．ただし，この積和演算コストは拡大率 2 の場合，縦横 2 pixel のステップで計算されることから，実質的に上記処理は 4 ピクセル分の出力する積和演算コストといえる．従って，上記トータル値を 4 で割った値である 49,000 step が図 3.4.4.の設定時における PDNN のハードウェア実装コストの見積もりといえる．

以上が PDNN を用いた超解像システムにおけるハードウェア実装時の大まかな見積もりを示すものである．ここで，2.4.4 項で述べた SRCNN の実装コストでは 1,700,704 step の積和演算と，6,273 pixel の中間メモリが最低限必要であったことを振り返ると，SR-PDNN のハードウェア実現コストは SRCNN に比べて大幅に削減していることがわかる．これらの比較に関する詳細は，第 4 章で述べるが，実装コストが大幅に削減している大きな原因として CNN から PDNN にアーキテクチャを変換したことが挙げられる．特に理由として大きいのは，CNN で実装コスト増大の原因となっていた受容野の拡大や次元数に応じた爆発的な特徴量の増加が PDNN のアーキテクチャでは発生しないことが挙げられる．また，拡大率に応じたステップ処理が計算量の大幅な削減にも寄与している．なお，実際のハードウェア実装には 3.4.4 項で述べたオーバーラップ部分に対する平均化処理が必要であるが，本処理はラインメモリの実装によって簡易に実現が可能であり，計算コストとしては非常に軽い．ラインメモリの実装は解像度に依存する部分もあるため今回のハードウェア実装の見積もりからは省いているが，影響は軽微である．

3.5　結言

本章では，本研究の課題である低遅延リアルタイムの実現に向けて，検討手法である SR-PDNN の設計について記述した．SR-PDNN は，従来の CNN ベースとは異なりパッチベースのニューラルネット（PDNN）によって構成されるため，ハードウェア実装に適した手法といえる．また，超解像の問題を改めて見直すことで不要な冗長を省き，適切な制約によって問題をシンプル化することで限られたコストで最大限の性能発揮を期待するものである．本章で述べた内容をまとめると，以下の通りとなる．

3.2 PDNN（Patch-based Deep Neural Network）の設計では，SR-PDNN のコアロジックとなるパッチベース型ニューラルネット（PDNN）に関する設計思想とアルゴリズムの詳細を記載した．具体的には，超解像問題を解く上で有効となる入出力定義から始まり，基底ベクトルの集合で表した出力層（復元部）の設計，復元情報を得るための入力層（特徴抽出部）の設計，特徴量から復元構成要素の係数を定める中間層（推定層）の設計を述べ，最後に PDNN の内部構成をまとめた．

3.3 学習による重み行列の最適化では, 3.2 節で述べた PDNN の学習方法について述べた．補間画素ピクセルを考慮した End-to-End 学習を行うことで，正解データに含まれるばらつきを軽減し，精度向上を目論むことを説明した．また，損失関数の設定や学習データセットの準備など，具体的な学習手順の方法についても記述した．

3.4 PDNN を用いた超解像システムでは，パッチベース処理である PDNN を用いて任意サイズに対して超解像処理を施す手法について説明した．特に，拡大率に応じたパッチ抽出や，パッチサイズの設定，さらにはオーバーラップ処理による確率的平均化効果について説明した．また，PDNN に基づいた超解像手法のハードウェア実装コストに関する簡易な見積もりも実施し，SRCNN と比較して大幅な削減が見込めることも述べた．

参考文献

[3-1] C. Dong, C.C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” European Conference on Computer Vision, vol. 8692, pp.184–199, Springer, 2014.

[3-2] J. Kim, J.K. Lee, and K.M. Lee, “Accurate image super-resolution using very deep convolutional networks,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1646–1654, 2016.

[3-3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.

[3-4] W. Shi, J. Caballero, F. Husz´ar, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1874–1883, 2016.

[3-5] 大谷，加藤，”4 並列の畳み込みニューラルネットワークを用いた超解像,” 電子情報通信学会論文誌, vol. J99-D no. 5, 2019.

[3-6] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

第 4 章 シミュレーションによる性能評価と解析

4.1 緒言

本章では，SR-PDNN の有効性を検証すべく，実際の画像を用いた数値実験に基づく復元性能評価とCNNベースの手法と比較したコストパフォーマンスについて述べる．また，検討手法の考察を深めるため，パッチ抽出時におけるスライド幅設置による補間画素位置特定に対する影響や，基底ベクトル数に応じた性能変化，さらには，学習データの違いによる復元性能と汎用性能の変化など，学習条件に応じた性能解析や考察についても述べる．

4.2 SR-PDNN の性能評価

4.2.1 復元性能

超解像システムとして最も重要な復元性能を先ず始めに評価する．表 4.2.1 は 3 章で述べたSR-PDNNを用いて実際の画像でシミュレーション状況を行うための実験条件の一覧である．本研究では SRCNN［4-1］を主たる比較対象として捉えているため，Dong らの論文と同様，拡大率としては 2, 3, 4 倍の 3 パターンを選択した．また，各拡大率に対応

表 4.2.1：SR-PDNN の実験条件

項目	拡大率: 2	拡大率: 3	拡大率: 4
入出力パッチサイズ (pixel)	6 × 6	9 × 9	12 × 12
特徴次元数 f_l	200	200	200
基底ベクトル数 f_h	800	800	800
学習用データセット	91 *images*	91 *images*	91 *images*
学習用データのパッチ数 (paired patches)	4,916,516	2,087,888	1,119,188
学習回数 (epochs)	5000	5000	5000
学習データ生成時のステップ幅 (pixel)	2	3	4
復元処理時のステップ幅 (pixel)	2	3	4

する入出力パッチサイズとしては 3.4.4 項の議論に基づき，拡大率の 3 倍である 6×6，9×9，12×12 に設定した．次に，ネットワーク規模を決定するパラメータである特徴次元数 f_l は 200 に，基底ベクトル数 f_h は 800 に設定した．さらに，SR-PDNN を学習する上で必要となる学習用データセットについては SRCNN（9-1-5）や ScSR［4-2］でも使われている 91 images と呼ばれるデータセットを使用した．ここで，学習データセットについてより詳しく説明すると，図 4.2.1 で示すように，91 images から作成された HR 画像と LR 画像のペア画像に対して，各拡大率に対応するパッサイズの画像を切り出したものを学習データとして利用した．ただし，学習データのパッチ画像切り出し時における水平・垂直方向のステップ幅は対象の拡大率と同じ値｛2, 3, 4｝とし，復元処理におけるステップ幅もそれぞれ同じ値とした．また，学習データ生成時には 90 度，180 度，270 度の回転によるデータ水増しも行った．結果として 91images から拡大率 2 の場合は 4,916,516 組，拡大率 3 の場合は 2,087,888 組，拡大率 4 の場合は 1,119,188 組のペアとなるパッチ画像が学習セットとしてそれぞれ生成された．

表 4.2.2 は本実験で比較する手法の学習条件を示している．本実験の比較では入力でかつ LR 画像の生成に使用している Bicubic 補間に加え，参考論文中で SR-PDNN と同じく 91 images を学習に利用している学習型超解像である ScSR と SRCNN（9-1-5）を選択した．ただし，拡大率 2 の ScSR については，著者らが提供するコードに付随して提供された学習済みパラメータ（ ScSR の場合は辞書に相当 ）を利用したが，拡大率 3 と 4 の学習済みパラメータについては著者提供コードには含まれていなかったため，著者コードを利用して辞書生成を行った．また，SRCNN（9-1-5）に関しては，著者らが提供した学習済みのモデルを使用し，参考論文の実験との整合性を担保した．

表 4.2.2：復元性能評価の比較対象

比較手法	拡大率: 2	拡大率: 3	拡大率: 4
Bicubic	–	–	–
ScSR	91 *images*（著者提供）	91 *images*（学習生成）	91 *images*（学習生成）
SRCNN（9-1-5）	91 *images*（著者提供）	91 *images*（著者提供）	91 *images*（著者提供）

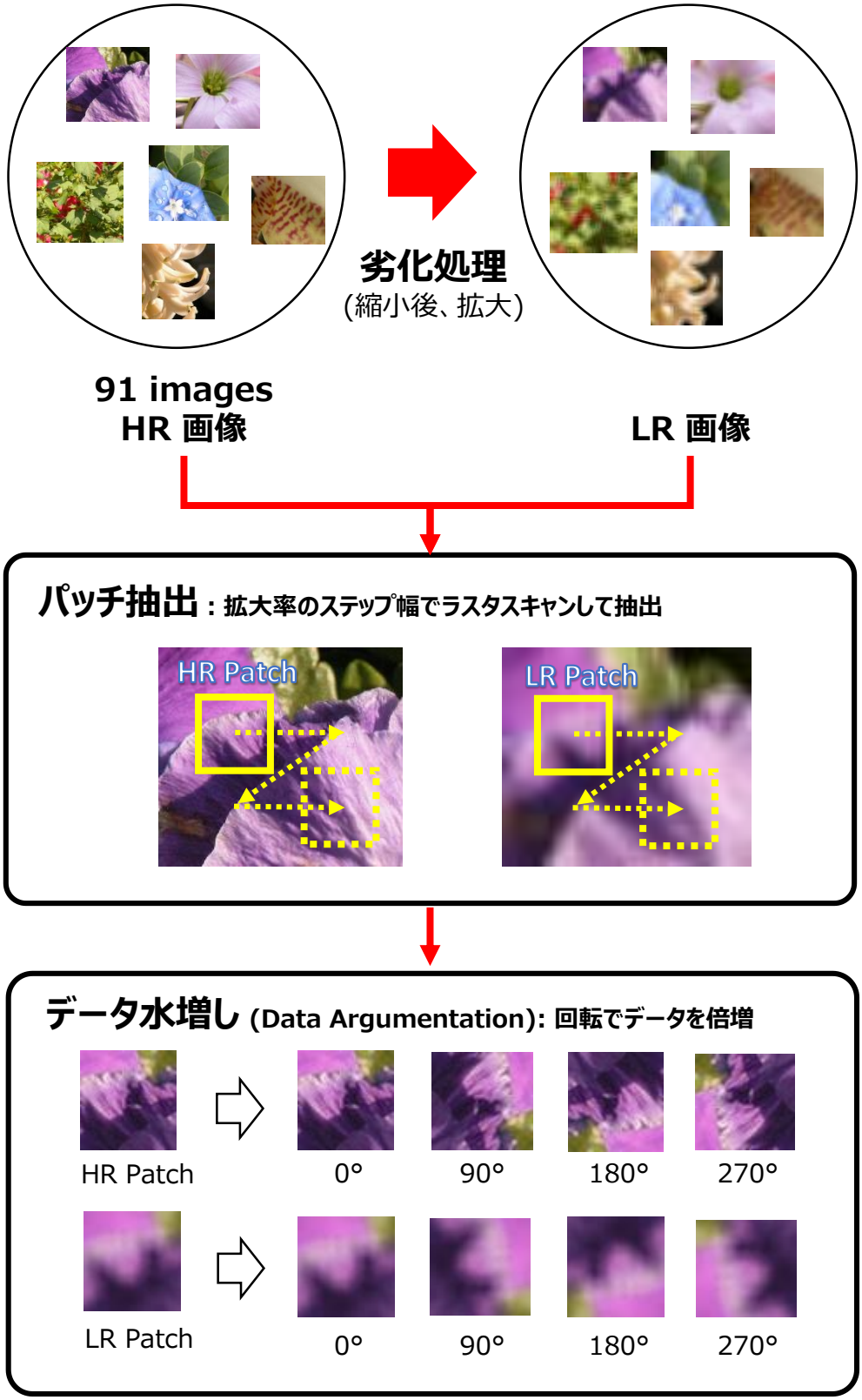
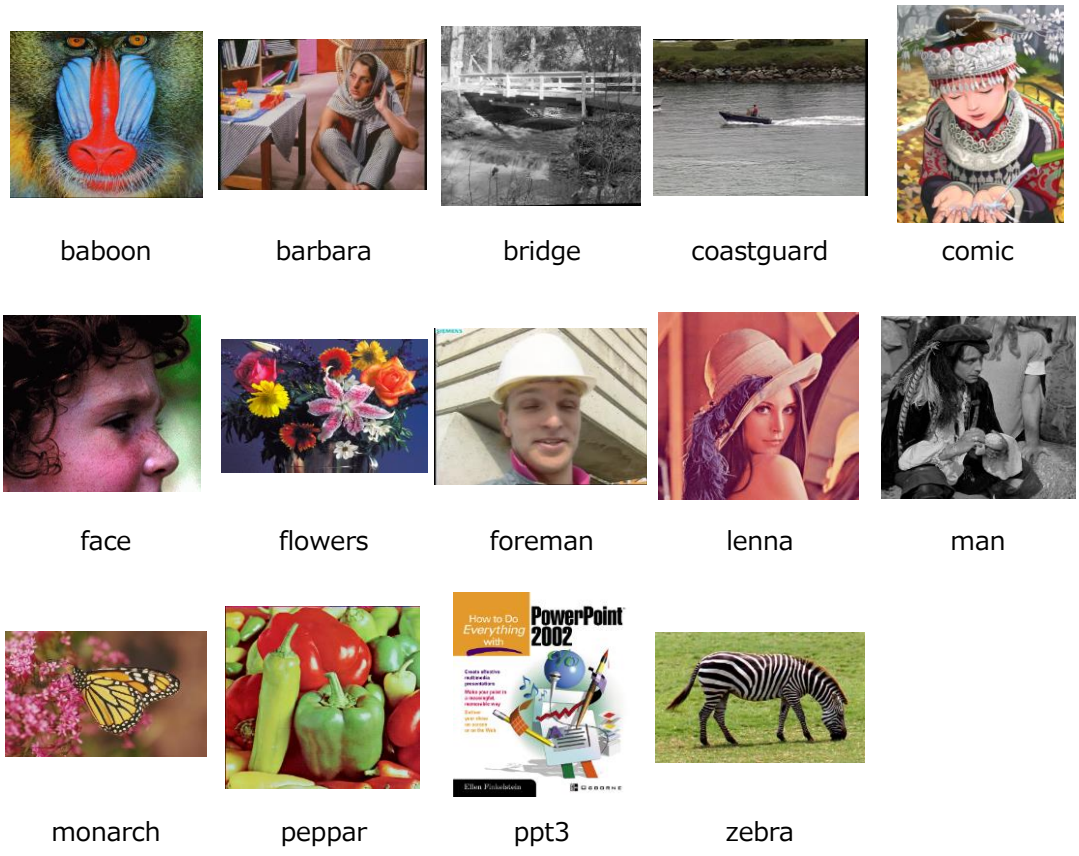


図 4.2.1： 学習データセットの生成手順

baby　bird　butterfly　head　woman

図 4.2.2：Set5



baboon　barbara　bridge　coastguard　comic

face　flowers　foreman　lenna　man

monarch　peppar　ppt3　zebra

図 4.2.3：Set14

評価画像に関しては 図 4.2.2 及び，図 4.2.3 で示した学習型超解像分野では標準的に古くから使用されている Set5 [4-3], Set14 [4-4] を本実験では選択した．理由はこれらの評価画像での実験結果が SRCNN や ScSR の論文にも結果が記載されており，公平性の高い評価が実現できるからである．なお，表 4.2.3 及び 表 4.2.4 は Set5 と Set14 の各画像のサイズと特筆すべき画像の特徴を示したものである．評価画像である Set5, Set14 には主に自然画を中心とした画像が含まれているが，ppt3 や comic など一部人工的な画

表 4.2.3：Set5 画像サイズ

画像名	横サイズ	縦サイズ	備考
baby	512	512	
bird	288	288	
butterfly	256	256	Set14 monarchの一部
head	280	280	Set14 face と同画像
woman	228	344	
baby	512	512	

表 4.2.4：Set14 画像サイズ

画像名	横サイズ	縦サイズ	備考
baboon	500	480	
barbara	720	576	
bridge	512	512	
coastguard	352	288	
comic	250	361	エッジの鋭い非自然画
face	276	276	
flower	500	362	
foreman	352	288	
lenna	512	512	
man	512	512	
monarch	768	512	
peppar	512	512	
ppt3	529	656	エッジの鋭い非自然画
zebra	586	391	

像が含まれている点や，head（face）には目立つノイズが含まれている点は今後評価を行う上で，データセットの特性として留意する必要がある．なお，画像サイズには統一性が無いが，実際の評価では各拡大率の値で割り切れない場合は予めトリミングしている．また，Set5, Se14 は RGB カラー画像であるが，実際の処理では YCbCr 画像に変換した後，Y 成分に対してのみ超解像処理を施す処理を前提としている．これらの制約はいずれも SRCNN, ScSR の実験条件と同じである．

表 4.2.5 から 表 4.2.7 は Set5 に対する復元性能の比較結果を示している．表に示した数値は式（3.3.4）で示した PSNR（単位：dB）であり，値が大きいほど正解画像である HR 画像に近いことを示している．また，表中の太文字は対象画像，もしくは平均値において最も高い数値であることを示している．つまり，提案手法である SR-PDNN は同一画像セットを用いて学習された ScSR，SRCNN（9-1-5）と比較して，評価画像セット Set5 において最も高い PSNR が得られた．これは他の手法より，もっとも正解画像に近い画像が出力されたことを示している．しかもその結論は倍率 2，3，4 のいずれの場合でも同様であり，定量的には 約 0.6 dB ~ 0.7 dB 程度の性能向上となった．この性能向上の度合いは，ScSR から SRCNN への性能向上幅が約 0.1 dB ~ 0.5 dB 程度であったことを考慮すると，高い性能向上が実現しているといえる．

表 4.2.5：Set5 復元性能比較（ 拡大率: 2) [単位 dB]

Set5 Scale: x2	Bicubic	ScSR	SRCNN (9-1-5)	SR-PDNN ($f_h = 800$)
baby	37.07	38.45	38.30	**38.53**
bird	36.81	40.56	40.64	**41.38**
butterfly	27.43	31.31	32.20	**33.48**
head	34.86	35.73	35.64	**35.77**
woman	32.14	34.95	34.94	**35.64**
average	33.66	36.20	36.34	**36.96**

表 4.2.6：Set5 復元性能比較（ 拡大率 ：3) [単位 dB]

Set5 Scale: x3	Bicubic	ScSR	SRCNN (9-1-5)	SR-PDNN ($f_h = 800$)
baby	33.91	35.02	35.01	**35.28**
bird	32.57	34.35	34.90	**35.84**
butterfly	24.04	26.22	27.58	**28.67**
head	32.88	33.56	33.55	**33.80**
woman	28.56	30.33	30.91	**31.84**
average	30.39	31.90	32.39	**33.09**

表 4.2.7：Set5 復元性能比較（ 拡大率 ：4) [単位 dB]

Set5 Scale: x4	Bicubic	ScSR	SRCNN (9-1-5)	SR-PDNN ($f_h = 800$)
baby	31.78	32.81	32.98	**33.30**
bird	30.18	31.55	31.97	**32.89**
butterfly	22.10	23.63	25.06	**25.79**
head	31.59	32.16	32.19	**32.52**
woman	26.46	27.66	28.20	**29.24**
average	28.42	29.56	30.08	**30.75**

表 4.2.8：Set14 復元結果（ 拡大率: 2 ）[単位 dB]

Set14 Scale: x2	Bicubic	ScSR	SRCNN (9-1-5)	SR-PDNN ($f_h = 800$)
baboon	24.86	25.59	25.62	**25.74**
barbara	28.00	**28.70**	28.59	28.64
bridge	26.58	27.67	27.70	**27.91**
coastguard	29.12	30.58	30.49	**30.73**
comic	26.02	27.99	28.27	**28.80**
face	34.83	35.71	35.62	**35.74**
flowers	30.37	32.72	33.03	**33.55**
foreman	34.14	36.91	36.23	**37.05**
lenna	34.70	36.48	36.50	**36.72**
man	29.25	30.69	30.82	**31.07**
monarch	32.94	36.52	37.18	**38.29**
pepper	34.95	36.73	36.73	**37.08**
ppt3	26.87	29.52	30.40	**31.01**
zebra	30.63	33.37	33.29	**33.96**
average	30.23	32.08	32.18	**32.59**

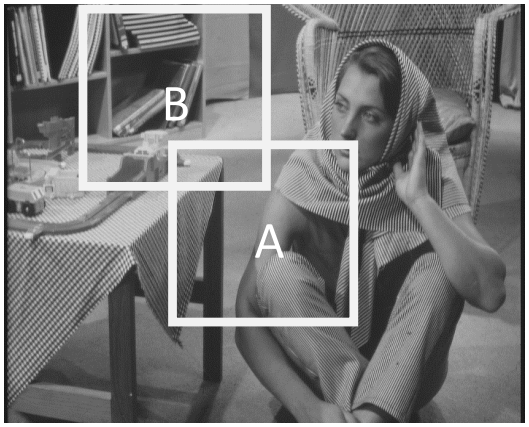
表 4.2.9：Set14 復元結果（ 拡大率: 3 ）[単位 dB]

Set14 Scale: x3	Bicubic	ScSR	SRCNN (9-1-5)	SR-PDNN ($f_h = 800$)
baboon	23.21	23.54	23.60	**23.69**
barbara	26.25	**26.70**	26.66	26.58
bridge	24.40	25.02	25.07	**25.31**
coastguard	26.55	27.18	27.20	**27.34**
comic	23.12	24.04	24.39	**24.71**
face	32.82	33.52	33.58	**33.79**
flowers	27.23	28.51	28.97	**29.44**
foreman	31.16	33.19	33.40	**34.41**
lenna	31.68	33.04	33.39	**33.77**
man	27.01	27.91	28.18	**28.46**
monarch	29.43	31.30	32.39	**33.38**
pepper	32.38	33.81	34.34	**34.90**
ppt3	23.71	25.06	26.02	**26.75**
zebra	26.63	28.38	28.87	**29.64**
average	27.54	28.66	29.00	**29.44**

表 4.2.10：Set14 復元結果（ 拡大率: 4) [単位 dB]

Set14 Scale: x4	Bicubic	ScSR	SRCNN (9-1-5)	SR-PDNN ($f_h = 800$)
baboon	22.44	22.66	22.70	**22.77**
barbara	25.15	25.57	25.70	**25.87**
bridge	23.15	23.58	23.65	**23.91**
coastguard	25.48	25.65	25.94	**26.07**
comic	21.69	22.28	22.53	**22.80**
face	31.55	32.09	32.12	**32.51**
flowers	25.52	26.41	26.84	**27.30**
foreman	29.38	30.45	31.46	**32.49**
lenna	29.83	30.81	31.20	**31.63**
man	25.70	26.38	26.65	**26.96**
monarch	27.46	28.80	29.89	**30.48**
pepper	30.59	31.70	32.34	**33.12**
ppt3	21.98	22.71	23.84	**24.48**
zebra	24.08	25.38	25.97	**26.55**
average	26.00	26.75	27.20	**27.64**

次に，表 4.2.8 から 表 4.2.10 に Set14 に対する復元性能の比較結果を示す．Set14 の場合も Set5 と同様に PSNR の平均値では提案手法である SR-PDNN の復元性能が最も高いという結果になっており，定量的には約 0.4 dB 程度 SRCNN よりも数値が高くなっていることが確認できる．ただし，例外として barbara では ScSR が最も高い数値となっている．図 4.2.4 は 拡大率が 3 の場合における barbara の ScSR と SR-PDNN の復元結果を示したものである．まず，barbara の画像特性としてはスカーフやストライプ柄のパンツ，テーブルクロスなどといった，規則正しいパターンが多く含まれているといった特徴がある．例えば，図中の矩形領域 A はそういった部分が多い箇所だが，実際に処理結果の画像を見ると SR-PDNN の結果は ScSR の結果に比べて微細な変化が抑えられている印象を受ける．この現象は Junstin ら [4-5] の報告にも在るとおり，MSE で定義された評価関数値を平均的に改善しようとした結果，テクスチャが鈍ってしまう現象として知られており，確率的に推定精度が低くなる高周波部分を敢えて平均化することで，結果的に大幅な外れ（損失関数の増加）を防いだ結果と解釈できる．ただし，この僅かなボケによる画質の差は単一画像でみたとき主観的には認識され難いものである．主観的に目立つ画質低下は，矩形領域 B に示す本の境界線等に現れる．当該エリアに見られるジャギーが SR-PDNN の結果では，ScSR と比較して明らかに低減していることが視覚的に分かりやすいため，総合的な主観品質は SR-PDNN のほうが良いと考えられる．このような客観指標である PSNR と主観の印象が必ずしも一致しないという点については，画像処理の分野では度々議論されることであるが，PSNR というものは主観評価結果と大まかな相関は示すことが出来るものの，必ずしも PSNR の大小が主観画質の結果と等しくなるわけではないということは念頭においておく必要がある．



barbara

(a) ScSR［領域Ａ］

(b) SR-PDNN［領域Ａ］

(c) ScSR［領域Ｂ］

(d) SR-PDNN［領域Ｂ］

図 4.2.4：barbara の結果比較（ 拡大率 ：3 ）

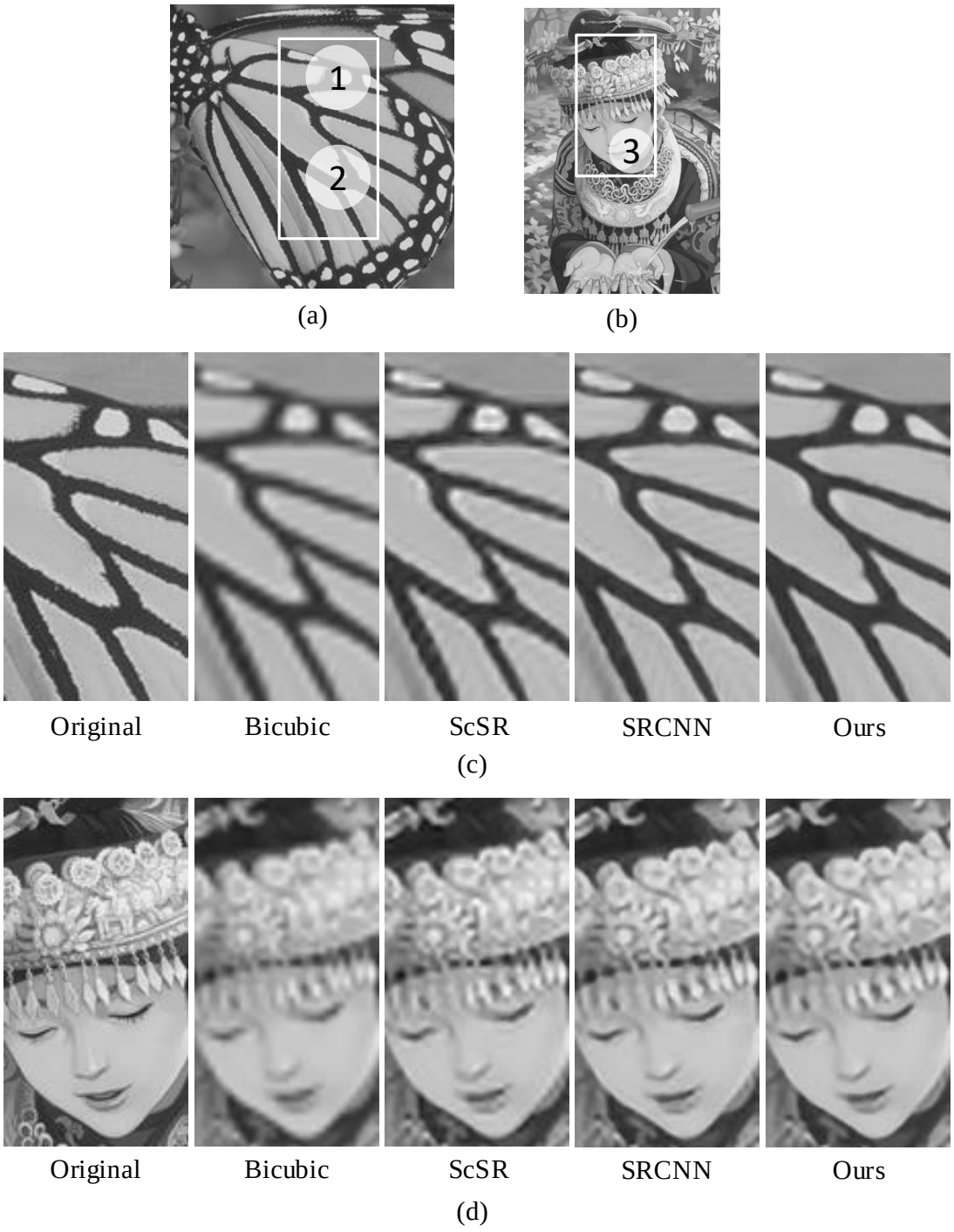


図 4.2.5： 復元結果画像の画質比較

図 4.2.5 は本実験の各手法の復元結果画像に対する主観的な違いを示すものである．ここで， 図中 (a) は Set5 に含まれる butterfly，(b) は Set14 に含まれる comic，(c) は拡大率 4 の場合における butterfly の各復元結果の一部を切り取ったものであり，(d) は拡大率 3 の場合における comic の各復元結果の一部を切り取ったものである．注

目点として，円領域 1 ではリンギングが，円領域 2, 3 ではジャギーが現れやすい領域である．なお，リンギングとは輪郭部分で生じるオーバーシュートやアンダーシュート現象である．各手法を比較すると，まず，入力の LR 画像である Bicubic には HR 画像である Originalには含まれていなかったぼけやジャギーが多く発生していることがわかる．また，ScSR, SRCNN, Ours（SR-PDNN）はいずれも LR 画像のボケを軽減する効果は見られるが，ScSR にはかなり多くのジャギーやリンギングが様々な輪郭線の部分を中心に残っていることがわかる．また，SRCNN の画質も ScSR に比べてジャギーの発生エリアは抑えられているものの，未だ多くの場所で残っている．対して，提案手法ではわずかにジャギーは残っているものの，殆どの箇所でジャギーやリンギングが軽減し，改善していることが明らかにわかる．

以上の結果から提案手法は 91 images という同一学習環境化において，従来の state of the art とされていた ScSR や SRCNN（9-1-5）を凌ぐ復元性能が得られていると評価出来る．また，その画質的な特徴としては，特に斜め線等で置きやすいジャギーを軽減され，特に人が見た場合に高い品質の画像が得られた．

4.2.2 コストパフォーマンス

前節では，91 images による学習という制約の中で提案手法のアーキテクチャが優れた復元性能を発揮できることを示した．本節では，SRCNN や VDSR [4-6] で用いられているCNN（Convolutional Neural Network）をベースとした手法と，提案手法である PDNN（Patch based Deep Neural Network）をベースとした手法との比較評価を行う．

表 4.2.11：比較対象である CNN 手法の学習条件

比較手法	拡大率: 2	拡大率: 3	拡大率: 4
Bicubic	–	–	–
SR-PDNN（f_h = 36）	91 *images* （学習生成）	91 *images* （学習生成）	91 *images* （学習生成）
SR-PDNN（f_h = 200）	91 *images* （学習生成）	91 *images* （学習生成）	91 *images* （学習生成）
SR-PDNN（f_h = 800）	91 *images* （学習生成）	91 *images* （学習生成）	91 *images* （学習生成）
SRCNN（9-1-5）	91 *images* （著者提供）	91 *images* （著者提供）	91 *images* （著者提供）
SRCNN（9-5-5）	***ImageNet*** （著者提供）	***ImageNet*** （著者提供）	***ImageNet*** （著者提供）
VDSR	**291 *images*** （著者提供）	**291 *images*** （著者提供）	**291 *images*** （著者提供）

表 4.2.12： ハードウェア実装コストと復元性能の比較（ 拡大率: 2 ）

比較手法 (Scale: 2)	ハードウェア実装コスト		復元性能				
	比率 (SRCNN 9-1-5 基準)		⊿PSNR [dB] (Bicubic 基準)				
	メモリ [pixel]	積和演算 [step]	Set5	Set14	B100	kodak	**Total**
SR-PDNN ($f_h = 36$)	10.59%	2.16%	2.88	2.03	1.58	1.73	**1.69**
SR-PDNN ($f_h = 200$)	16.97%	7.49%	3.22	2.29	1.80	2.01	**1.93**
SR-PDNN ($f_h = 800$)	40.33%	26.98%	3.30	2.36	1.86	2.07	**2.00**
SRCNN (9-1-5)	100.00%	100.00%	2.68	1.95	1.58	1.77	**1.69**
SRCNN (9-5-5)	244.18%	936.51%	3.00	2.22	1.80	1.96	**1.91**
VDSR	26619.58%	185980.12%	3.85	2.82	2.16	2.74	**2.38**

表 4.2.13： ハードウェア実装コストと復元性能の比較（ 拡大率: 3 ）

比較手法 (Scale: 3)	ハードウェア実装コスト		復元性能				
	比率 (SRCNN 9-1-5 基準)		⊿PSNR [dB] (Bicubic 基準)				
	メモリ [pixel]	積和演算 [step]	Set5	Set14	B100	kodak	**Total**
SR-PDNN ($f_h = 36$)	12.34%	3.62%	2.13	1.53	1.01	1.03	**1.10**
SR-PDNN ($f_h = 200$)	18.72%	9.97%	2.56	1.81	1.21	1.24	**1.32**
SR-PDNN ($f_h = 800$)	42.08%	33.18%	2.70	1.90	1.27	1.31	**1.39**
SRCNN (9-1-5)	100.00%	100.00%	2.00	1.46	1.00	1.00	**1.08**
SRCNN (9-5-5)	244.18%	936.51%	2.36	1.75	1.20	1.22	**1.30**
VDSR	26619.58%	185980.12%	3.36	2.24	1.54	1.76	**1.71**

表 4.2.14： ハードウェア実装コストと復元性能の比較（ 拡大率: 4 ）

比較手法 (Scale: 4)	ハードウェア実装コスト		復元性能				
	比率 (SRCNN 9-1-5 基準)		⊿PSNR [dB] (Bicubic 基準)				
	メモリ [pixel]	積和演算 [step]	Set5	Set14	B100	kodak	**Total**
SR-PDNN ($f_h = 36$)	14.79%	5.67%	1.81	1.25	0.80	0.80	**0.88**
SR-PDNN ($f_h = 200$)	21.18%	13.44%	2.22	1.56	0.97	0.98	**1.07**
SR-PDNN ($f_h = 800$)	44.53%	41.85%	2.33	1.64	1.03	1.03	**1.13**
SRCNN (9-1-5)	100.00%	100.00%	1.66	1.20	0.75	0.72	**0.82**
SRCNN (9-5-5)	244.18%	936.51%	2.06	1.50	0.94	0.96	**1.04**
VDSR	26619.58%	185980.12%	2.86	2.02	1.26	1.35	**1.41**

表 4.2.11 は本実験での比較対象となる学習条件をそれぞれ示している．まず，提案手法である SR-PDNN では，学習データとして，4.2.1 項と同様に 91 images を使用する．ただし，ネットワーク規模（すなわちハードウェア実装コスト）の変化に対する性能変化を見るため，ハイパーパラメータである基底ベクトル数 f_{h} に関しては 36, 200, 800 を選

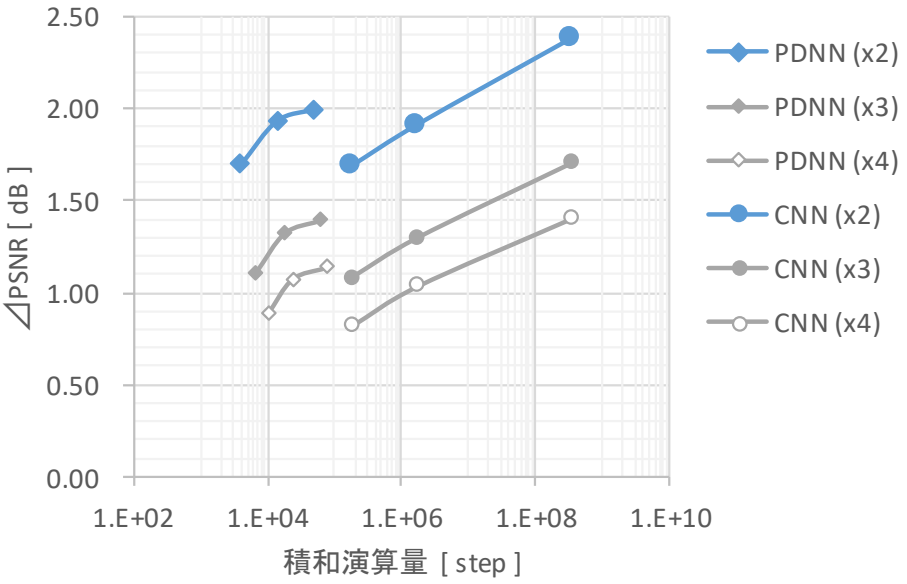


図 4.2.6：積和演算量に対する性能比較

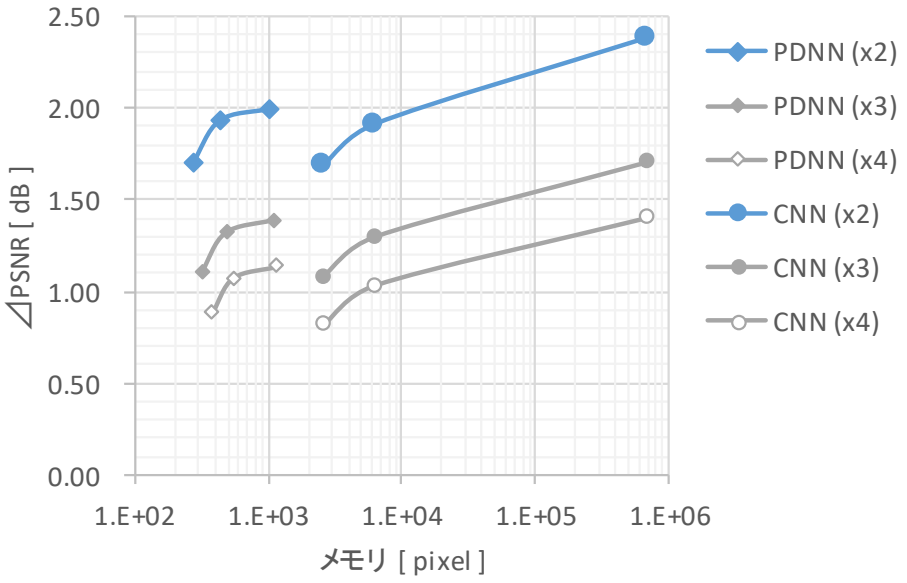


図 4.2.7：メモリコストに対する性能比較

択した. SRCNN の実験条件としては 9-1-5 モデルには 4.2.1 項と同様 91 images での学習結果を利用するが，9-5-5 モデルに関しては約 500 万個の ImageNet 画像から学習して得られたとされる Dong らが提供している学習済みのパラメータを使用した. VDSR については 291 images と呼ばれる 91 images に 200 枚の BSD（ Berkeley

Segmentation Dataset ）データセット [4-7] を利用したとされる著者提供パラメータを利用した. また, 評価データに関してもより幅広い画像評価を実現するために Set5, Set14 だけでなく B100 [4-7] とよばれる 100 枚のデータセットと, 24 枚の Kodak [4-8] データセットを加え，倍率 2 , 3, 4 について復元性能の比較実験を行った.

表 4.2.12 から 表 4.2.14 は各手法に対するハードウェア実装コストと復元性能の結果を纏めたものである．ここでハードウェア実装コストについては 2.4.4 項 及び 3.4.5 項にて計算手法の詳細は述べた通りであるが，ここでは比較を容易性にするため SRCNN（9-1-5）のハードウェア実装コストの見積もり値を 100% として比率で表示した．また，復元性能については入力である Bicubic 画像を基準とし，そこからの PSNR 上昇値をスコアとして記載した．ここで，Set5, Set14, B100, Kodak は各データセットでの平均値であり，Total とは全ての評価画像の平均値である．ここで，復元性能について各表を比較すると，概ね $f_h = 36$ のときが SRCNN（ 9-1-5 ） 相当の復元性能であり， $f_h = 200$ のときが SRCNN（9-5-5）相当であることがわかる. また, $f_h = 800$ では SRCNN（9-5-5）を凌ぐ復元性能が得られているが, VDSR には及んでいないことも分かる. しかしながら, VDSRの計算コストを見ると，メモリが SRCNN の約 2.6 万倍，積和演算量が約 18 万倍であることから文字通り桁違いのコストであり，本研究の目的からは外れていることが改めて認識できる.

図 4.2.6 及び 図 4.2.7 は積和演算量 及び メモリコストを横軸にし，Total の⊿PSNRを縦軸にした場合の結果を示している．このときコストパフォーマンスが高い状態というのは，低い実現コストで高い復元性能を持つ必要があることから，図中では左上に位置するほどコストパフォーマンスが高いことになる．つまり，PDNN 手法は明らかに CNN 手法と比べて低いコストで同等の性能を実現できており，コストパフォーマンスが高い手法であるということがわかる．より具体的には，表 4.2.12 より SRCNN（9-1-5）相当の復元性能を SR-PDNN で実現するには $f_h = 36$ 程度で十分であり，かつ，そのメモリコストは約 11 % ，積和演算量は約 2 % となる．これらの数値は，2.4.4 項の考察から考えても十分に実現性の高い規模まで縮小できていると捉えられる数値である.

ところで, 図 4.2.6 には提案手法のコストパフォーマンスが高いというほかに読み取れる部分が２つある．まず１つ目は CNN と PDNN の伸び代の違いである．CNN 手法のコストパフォーマンスは低いとはいえ，対数軸に対してほぼ比例する形状で性能が伸びているのに対し，PDNN 手法は頭打ちしているように見られる．実際，頭打ちしていることは 4.3.2 項で詳しく述べるが，この原因については入出力のパッチサイズを固定したことによる性能限界であると考えられる．一方で，CNN は層を深めるごとに受容野は拡がっていき性能を高めることができているが, 受容野の増大はパターンの増加に繋がるため, 結果的に処理は膨大となり計算量の観点では非効率になっている．次に２つ目は，拡大率に応じた復元限界の違いについてである．例えば，Set5 の VDSR に注目すると，拡大率 2 の場合で平均 3.85 dB 程度の向上が見込めるが，表 4.2.5 より入力の平均 PSNR は 33.66 dB であるから，復元後の平均 PSNR は 37.46 dB になる．同様に，拡大率 3 の場合を計算すると 33.75 dB であり，拡大率 4 の場合を計算すると 31.28 dB となる．つまり，現状の条件下では，拡大率が高い画像の復元結果が拡大率の低い画像の復元結果を超えることはないということが確認できる．なお，本研究では従来の Full-HD 画質を

4K , 8K 程度に拡大することを主たる目的と捉えると，２倍拡大が最もニーズが高く，４倍拡大まで対応していればよいと考えられる．上記視点から拡大率が 2 倍のときにおける画質の違いを示したのが 図 4.2.8 と図 4.2.9 である．ここで，図 4.2.8 では テクスチャ復元性能の比較として, Kodak データセットにある kodim18 画像の女性の顔の部分を選択し，図 4.2.9 ではジャギーが発生しやすい斜め線の構造物として kodim08 画像の屋根の一部を選択した．まず図 4.2.8 の図で復元されたテクスチャを比較すると, Bicubic と復元処理後の結果，及び，復元処理後の結果と Ground Truth の違いは明確にわかるものの，各手法の処理結果の比較としては殆ど差が分からないレベルであり，例え VDSR を用いたとしてもやはり女性の瞼はくすんだままであることが分かる．ではこの PSNR の違いはどこから現れているのかというと，図 4.2.9 のようなジャギーが目立つ斜め線画像で違いが出てきているのである．4.2.1 項でも述べたとおり，Ours 36, Ours 200 はそれぞれ SRCNN 9-1-5, SRCNN 9-5-5 相当のPSNRを出しており, 徐々に画質も改善され，Ours 800 ではほとんどのジャギーは軽減されているが，VDSR を確認すると完全といっていいレベルでのジャギー除去がされているように見受けられる．しかしながら，この差を得るためには SRCNN 9-1-5 比で約 18 万倍，Ours800 比であれば約 69 万倍のコストが掛かることになる．

以上の結果，及び，考察から，SR-PDNN はリアルタイム処理が可能なコストでありながらも, VDSR と比較しても十分に実用性のある復元性能をもった超解像処理であると判断出来る．

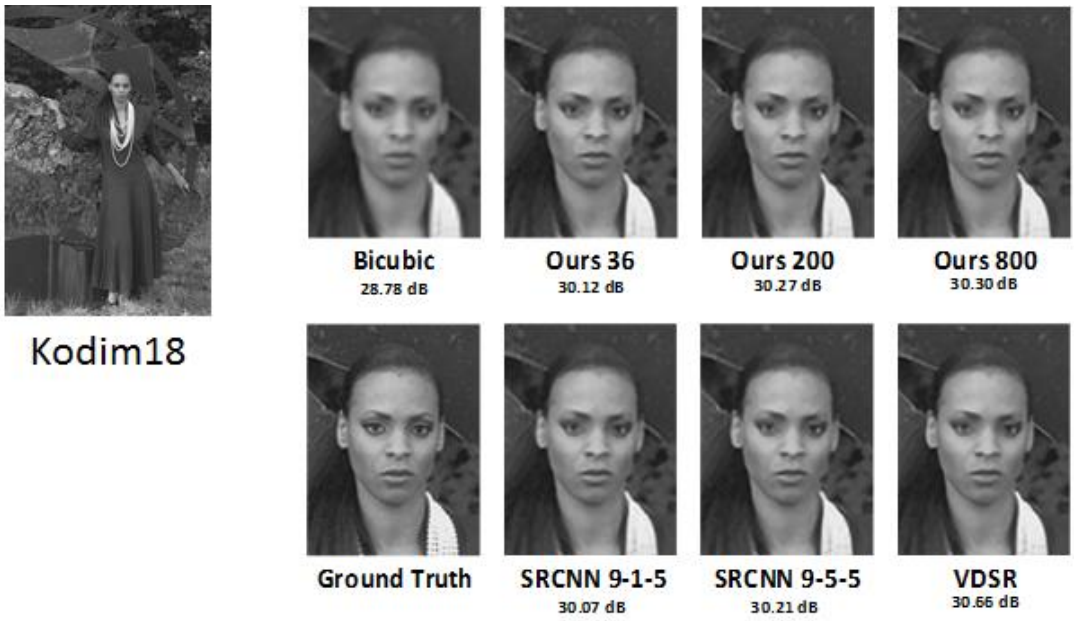


図 4.2.8： 拡大率: 2 における画質の違い①

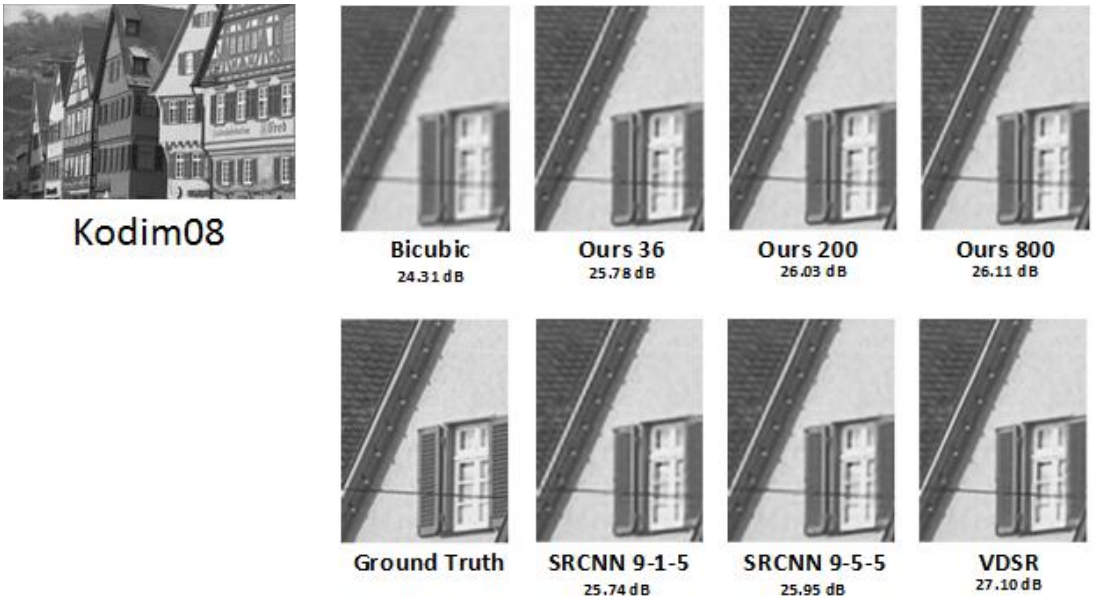


図 4.2.9： 拡大率: 2 における画質の違い②

4.3　学習条件に応じた性能解析

SR-PDNN が従来手法と比較してもコストパフォーマンスに優れた手法であることを 4.2 節では述べた．本節では，SR-PDNN が性能を発揮する条件や更なるコストパフォーマンス向上に向けた知見を述べる．

4.3.1　補間画素位置の特定による性能変化

本節では 3.3.1 項でも述べた補完画素位置と PDNN の各ニューロンとの一致性に対する効果について，数値実験による検証結果を述べる．表 4.3.1 は本実験のパラメータを示す．本実験では，拡大率 3 の PDNN を対象とし，学習データ生成時のステップ幅を 1, 2, 3 の 3 通りに設定した．また，復元処理時のステップ幅はパッチサイズ幅を最大サイズとし，1, 2, 3, 4, 5, 6, 7, 8, 9 の 9 通りとした．つまり，実際の学習としては，$3 \times 9 = 27$ 通りの実験を行った．ここで，ステップ幅 9 とは，パッチのオーバーラップなしであることを意味し，ステップ幅 1 とは 1 ピクセルごとに PDNN 対象画素がシフトしていく状態を意味する．

表 4.3.1：実験条件（拡大率: 3）

項目	Scale: 3
入出力パッチサイズ	9×9
特徴次元数 f_l	200
基底ベクトル数 f_h	800
学習用データセット	91 *images*
学習データ生成時のステップ幅	1,2,3
復元処理時のステップ幅	1,2,3,4,5,6,7,8,9

表 4.3.2 及び，表 4.3.4 は学習後に Set5，Set14 で評価した結果の PSNR を示している．各表を見れば，学習データ生成時のステップ幅（step for dataset creation）が拡大率と同じ 3 で，かつ，復元処理時のステップ幅もまた拡大率と同じ 3 のとき，最も高い PSNR が得られることが確認できる．つまり，3.3.1 項と 3.4.4 項で述べた対象パッチの周囲 8 近傍の結果を重ね合わせた場合が最も PSNR 上での復元成果が高くなることが確認できた．

表 4.3.2：Set5 での PSNR 比較結果［単位：dB］

		step for dataset creation		
		1	2	3
step for restoration	1	32.98	32.93	30.77
	2	32.97	32.91	30.68
	3	33.01	32.94	**33.09**
	4	32.92	32.85	29.47
	5	32.85	32.79	29.92
	6	32.86	32.82	32.94
	7	32.73	32.67	28.12
	8	32.64	32.57	28.16
	9	32.62	32.55	32.75

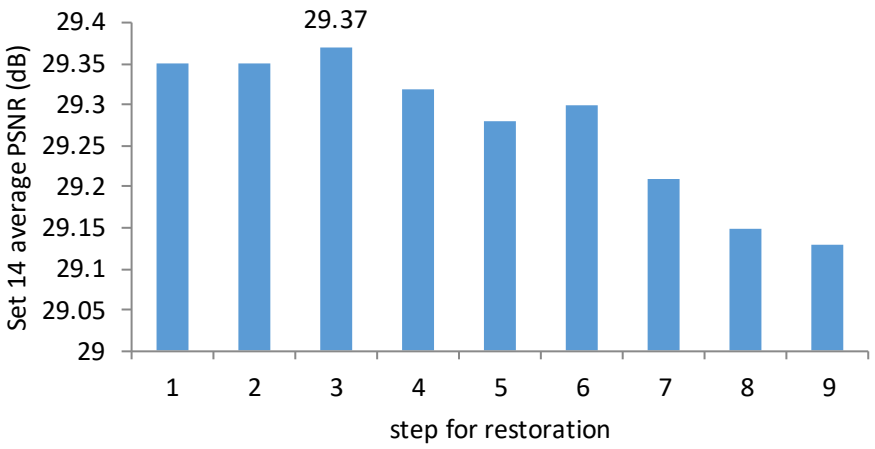
表 4.3.3：Set14 での PSNR 比較結果［単位：dB］

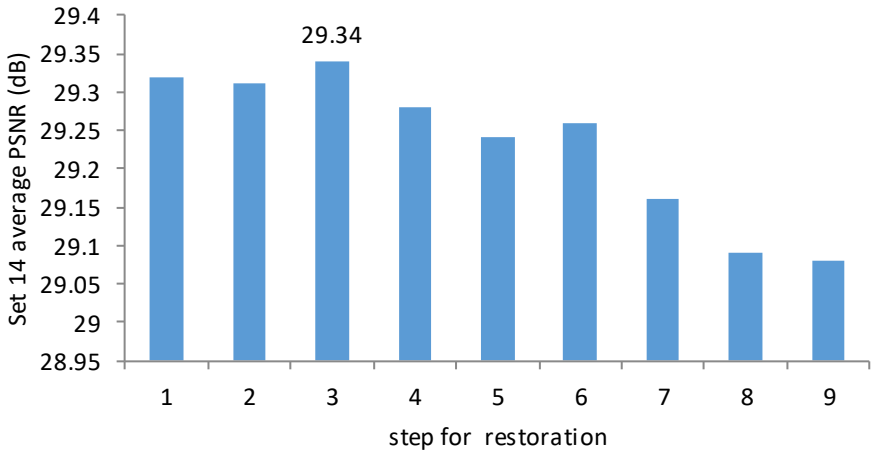
		step for dataset creation		
		1	2	3
step for restoration	1	29.35	29.32	27.72
	2	29.35	29.31	27.66
	3	29.37	29.34	**29.44**
	4	29.32	29.28	26.8
	5	29.28	29.24	27.12
	6	29.3	29.26	29.36
	7	29.21	29.16	25.84
	8	29.15	29.09	25.75
	9	29.13	29.08	29.22

次に，図 4.3.1 及び 図 4.3.2 は 表 4.3.2 と表 4.3.3 の結果に対して，学習データセット生成時のステップ幅ごとの結果を棒グラフに纏めたものである．ここで，復元性能が最も高いステップ幅が 3 の場合に注目すると，復元時のステップ幅が 3 の倍数の場合に限って，周期的に PSNR の数値が高くなっていることが確認できる．この原因は図 4.3.3 で示すように，9 × 9 サイズで切り出されたパッチ画像は 3 × 3 パターンで表される 9 つのブロックで構成されることに起因する．つまり，復元時のステップ幅が 3 の時は，オーバーラップと平均化処理はすぐ隣の 3 × 3 サイズのブロックとの位置関係が丁度合わさった時であり，復元時のステップ幅が 6 の時は，1 つ飛ばしの 3 × 3 ブロックが重なり，復元ステップ幅が 9 のときは 2 個飛ばしで位置関係が一致したと考えられる．

また，学習データ生成時のステップ幅が 3 でかつ，復元処理時のステップ幅が 3, 6, 9 以外の場合を見ると，総じて PSNR の値が低く，かつ，ステップ幅が大きくなるほど減少していく傾向にある．これは，学習時に各ニューロンの位置が整えられているのにも関わらず，復元時にはその位置関係を無視した適用をしたためと解釈できる．また，復元時のステップ幅が大きくなるにつれて徐々に PSNR が低下しているのは，オーバーラップ面積の低下によって， 3.4.5 項でも述べた平均化効果が薄れていくためと考えられる．また，学習データ生成時のステップ幅が 1 もしくは 2 の場合を見ると，学習データ生成時のステップ幅が 3 の場合と比較して，ピーク性能は低いものの全体的に高い性能を持っていることが分かる．表 4.3.4 は図 4.3.3 の画像に対して右方向に走査した場合における各パッチ画像の左上画素位置を示したものである．図の通り，学習データ生成時のステップ幅が 1 や 2 の場合は，抽出されたパッチ画像の補間画素パターンは固定されず，すべてのパターンが各ニューロンに対して入力されるということがわかる．従って，学習データとしては汎用性の高い学習が成されるため，結果的にどの位置で復元時のパッチを切り抜いたとしても，それなりの復元性能が得られたと解釈できる．しかしながら，最高性能を出すという面では，正解画像の余分な変動を除外した，学習データ生成時のステップ幅が拡大率と同じで，かつ，復元処理時のステップ幅も拡大率と同じ場合が最も高い性能を示すということは，本研究の狙い通りの結果であるといえる．なお，学習データ生成時のステップ幅が 1 や 2 の場合においても復元処理時のステップ幅が増加すると PSNR が減少していく方向にある点については，平均化処理による効果が大きいと解釈できる．また，復元処理時のステップ幅が 3 の倍数のときに若干高い PSNR が得られている理由は，学習データは一致していないが，復元処理としての補間画素パターンの位相が一致しているためと捉えることができる．

以上の実験により，各ニューロンの入力に対して補間画素位置を固定することで，推定精度を上げるという当初の目的の 1 つは達成できていることが確認できた．





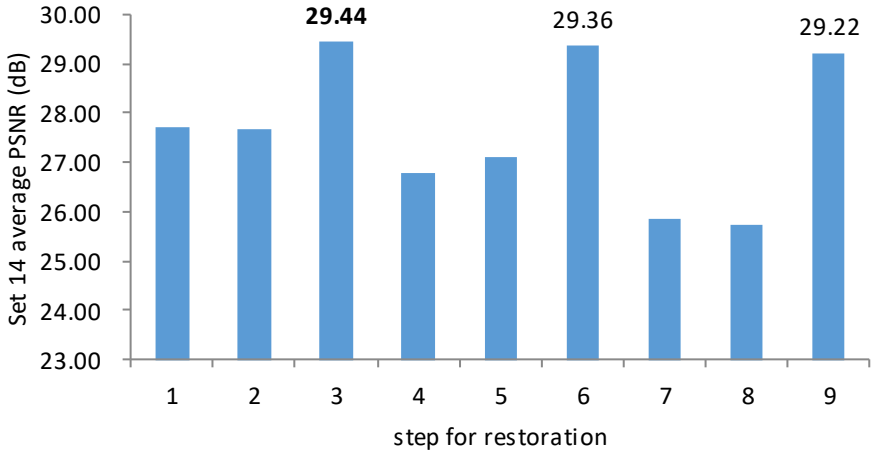


図 4.3.2：Set 14 での PSNR 比較結果
（上から順に，学習データ生成時のステップ幅は 1，2，3）

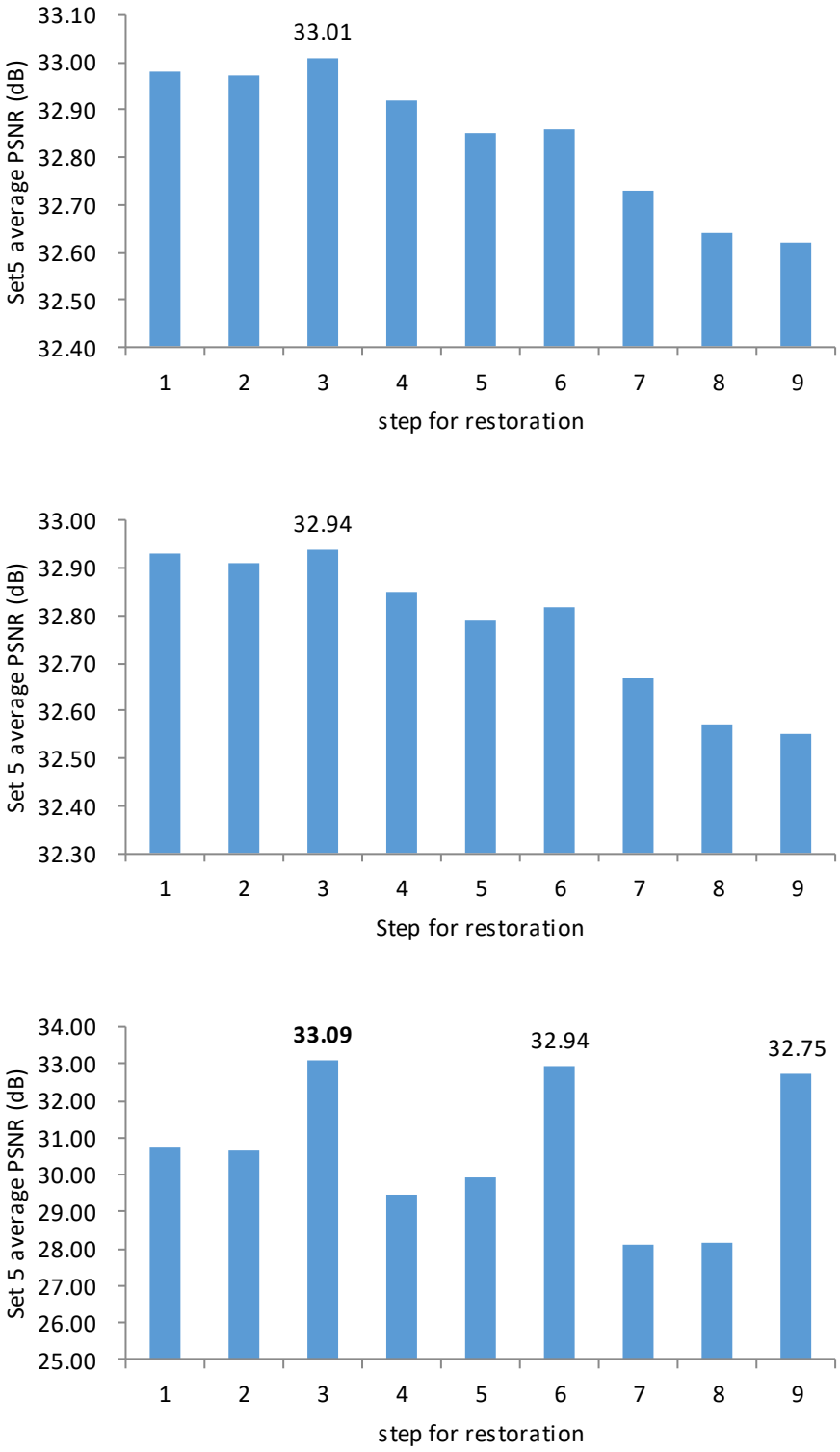


図 4.3.1：Set5 での PSNR 比較結果
（上から順に，学習データ生成時のステップ幅は 1，2，3）

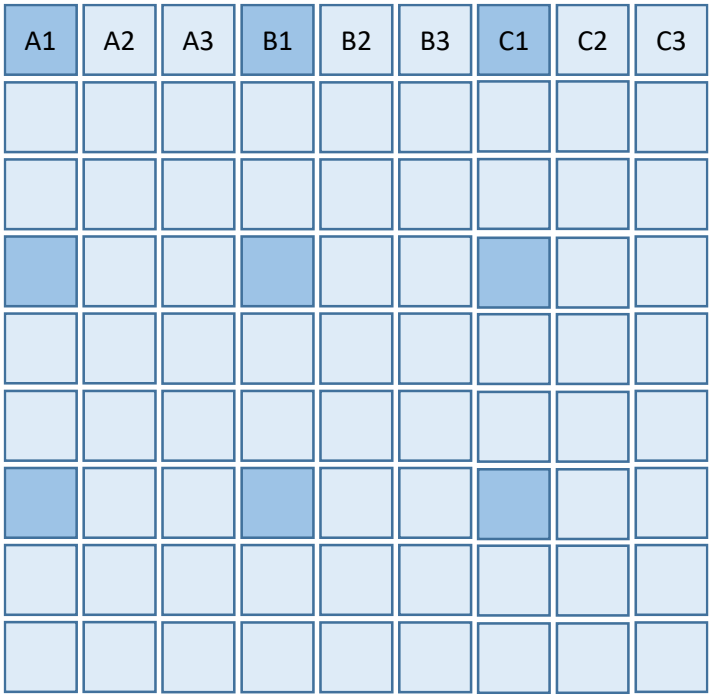


図 4.3.3： 画像に含まれる補間画素のパターン

表 4.3.4 :学習データ生成時のステップ数の違いによる補間画素パターンの変化

Image Data	A1	A2	A3	B1	B2	B3	C1	C2	C3
Step for dataset creation = 1	A1	A2	A3	B1	B2	B3	C1	C2	C3
Step for dataset creation = 2	A1	A3	B2	C1	C3	D2	E1	E3	F2
Step for dataset creation = 3	A1	B1	C1	D1	E1	F1	G1	H1	I1

4.3.2　基底ベクトル数の復元性能に対する影響

本節では SR-PDNN のハイパーパラメータである基底ベクトル数 f_h を変化させた場合における復元性能の変化を調べると共に，重み行列の学習結果を可視化し，SR-PDNN のネットワーク規模の最適性や性能限界，そして，コストパフォーマンスの向上について述べる．

表 4.3.5 は本実験の実験条件を示したものである．本実験では，4.2.1 項の拡大率 2 の場合を基準とし，$f_h = 18, 36, 72, 100, 200, 400, 600, 800, 1000$ の合計 9 パターンを学習し，性能比較を実施した．図 4.3.4 は基底ベクトル数を横軸に，Set14 の平均 PSNR を縦軸にした場合の実験結果を示している．結果より，基底ベクトルの増加によって，SR-PDNN の復元性能は対数曲線を描いて増加し，やがて頭打ちとなっている様子が伺える．また，図中に点線で示したとおり，SRCNN 9-1-5 モデル 及び， SRCNN 9-5-5 モデルに対応する性能に到達するポイントとしては，$f_h = 36$ 及び，$f_h = 200$ 付近であることが確認できる．表 4.3.6 は $f_h = 36, 200, 800$ の場合における性能比較を示しているが，概ね $f_h = 36$ が SRCNN 9-1-5 相当であり，また，$f_h = 200$ が SRCNN 9-5-5 相当であることが 4.2.2 項と同様，改めて確認できる．以上の結果から，SR-PDNN はハイパーパラメータで設定されるネットワーク規模によって性能は変化し，ネットワーク規模が大きいほど性能は向上するが，青天井というわけではなく，ある程度限界があることが観察された．

ここで，SR-PDNN の性能限界についてより詳しく解析するため，学習対象である重み行列 W_{filter}, W_{atom} についても更なる結果解析を行っていく．3.2.3 項で述べたとおり，重み行列 W_{filter} はパッチ入力画像 y の特徴量を抽出するために f_l 個の特徴抽出フィルターによって形成されていることから，各特長抽出フィルターをパッチ画像として表現すれば，可視化することができる．

表 4.3.5：実験条件（拡大率: 2）

項目	Scale: 2
入出力パッチサイズ	6 × 6
特徴次元数 f_l	200
基底ベクトル数 f_h	**18, 36, 72, 100, 200, 400, 600, 800, 1000**
学習用データセット	91 *images*
学習用データのパッチ数	4,916,516
学習データ生成時のステップ幅	2
復元処理時のステップ幅	2

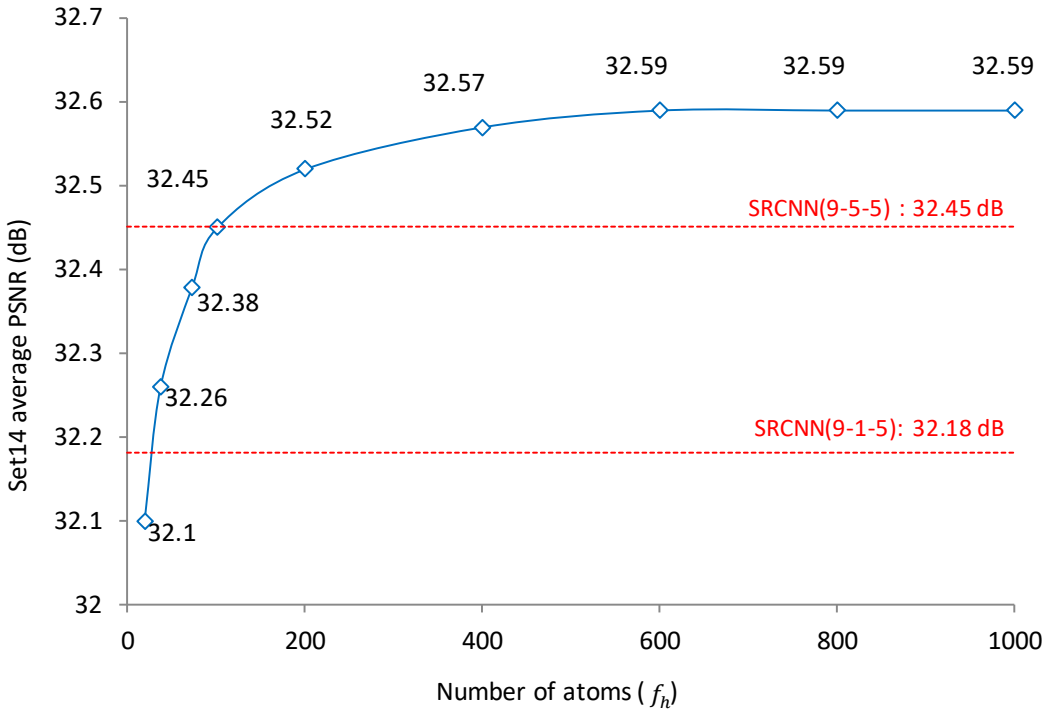


図 4.3.4： 基底ベクトルに応じた復元性能の変化

表 4.3.6 Set14 復元性能（PSNR）の比較

Set14 Scale: x2	Bicubic	ScSR	SRCNN (9-1-5)	SRCNN (9-5-5)	SR-PDNN ($f_h = 36$)	SR-PDNN ($f_h = 200$)	SR-PDNN ($f_h = 800$)
baboon	24.86	25.59	25.62	**25.74**	25.63	25.73	**25.74**
barbara	28.00	**28.70**	28.59	28.64	28.67	28.70	28.64
bridge	26.58	27.67	27.70	27.83	27.78	27.87	**27.91**
coastguard	29.12	30.58	30.49	30.83	30.66	30.71	**30.73**
comic	26.02	27.99	28.27	28.52	28.27	28.68	**28.80**
face	34.83	35.71	35.62	35.70	35.68	35.69	**35.74**
flowers	30.37	32.72	33.03	33.32	32.93	33.43	**33.55**
foreman	34.14	36.91	36.23	36.47	36.55	36.85	**37.05**
lenna	34.70	36.48	36.50	36.64	36.49	36.65	**36.72**
man	29.25	30.69	30.82	31.04	30.77	31.01	**31.07**
monarch	32.94	36.52	37.18	37.74	37.19	38.03	**38.29**
pepper	34.95	36.73	36.73	36.87	36.85	37.02	**37.08**
ppt3	26.87	29.52	30.40	**31.52**	30.13	30.83	31.01
zebra	30.63	33.37	33.29	33.49	33.84	33.88	**33.96**
average	30.23	32.08	32.18	32.45	32.26	32.52	32.59

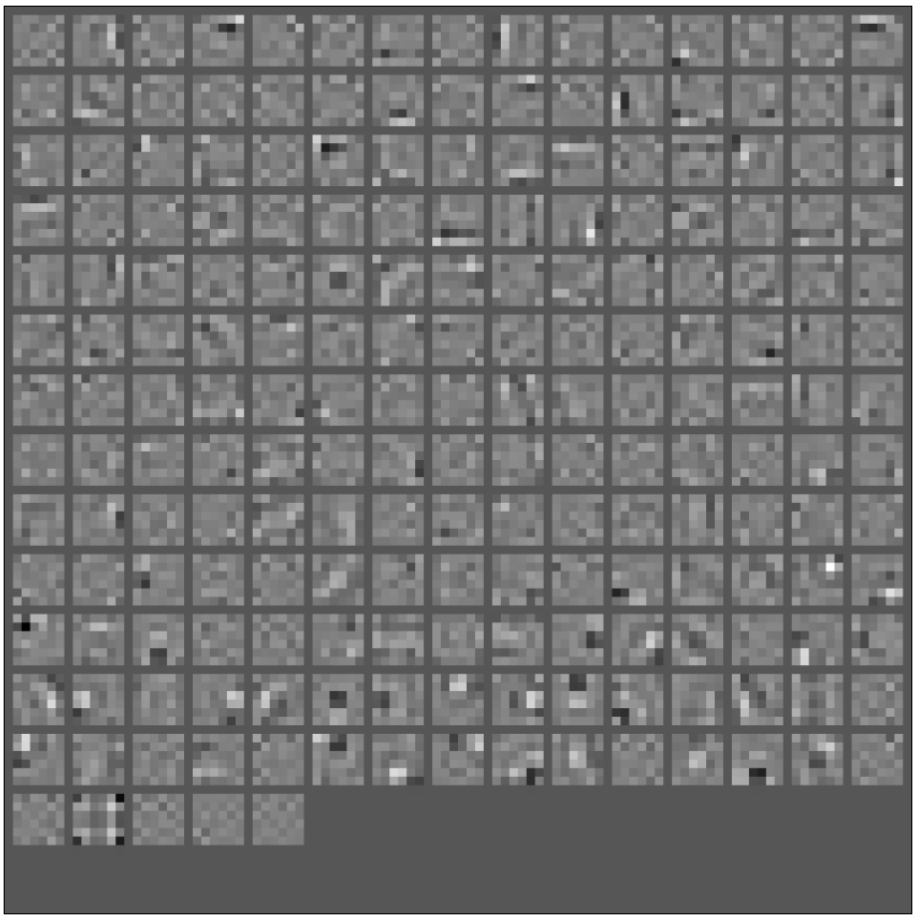


図 4.3.5： 特徴抽出フィルターの学習結果（$f_l = 200$）

図 4.3.5 は本実験において，$f_l = 200, f_h = 200$ の場合における学習済みの W_{filter} に含まれる全ての特徴抽出フィルターを可視化したものである. 図より機械学習以前の従来，アルゴリズム設計者が設定してきたラプラシアンフィルターとは明らかに異なる複雑なフィルターが構成されていることがわかる．機械学習という最適化手法によって，人が考え付くフィルター形状とは異なるが，全体として復元性能が高まるように最適化された結果と読み取れる．

一方で，W_{atom} もまた入出力画像サイズと同じ大きさの基底ベクトルの集合で表されるため，W_{filter} と同様に可視化が出来る．図 4.3.6 は図 4.3.5 と同条件における W_{atom} を基底ベクトルの絶対値が大きい順に並べ，可視化したものである．明らかに上部に並ぶ基底ベクトルは高周波成分を生成する上で大きな役割を果たし，下に行くにつれて振幅の弱い，つまり，出力への寄与が小さいベクトルが並んでいることが分かる．ここで，基底ベクトルの絶対値の総和（ atom size ） に着目し，$f_h = 36, 200, 800$ の場合における各 atom size の分布をまとめると図 4.3.7 となる．図 4.3.7 では，図 4.3.6 と同様に絶対値の総和が大きい順に並び替えているが，ここで注目すべきは $f_h = 36, 200$ では，それぞれ上限いっぱいまで埋まっていた有効な基底ベクトル（基底ベクトルの絶対値の総和がゼロ

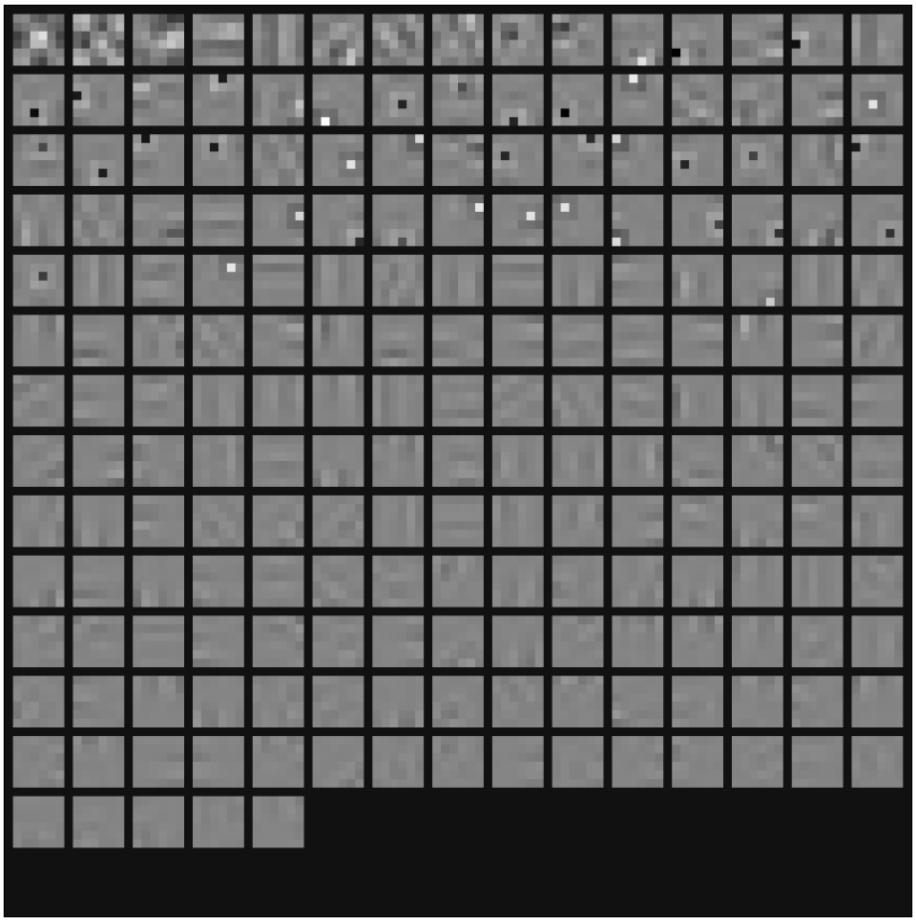


図 4.3.6：基底ベクトルの学習結果（$f_h = 200$）

ではない基底ベクトル）が，$f_h = 800$ では 600 個程度の基底ベクトルしか有効に働いていないという点である．これは，図 4.3.4 に示した基底ベクトルに対する復元性能変化とも合致が取れている．

つまり本現象は，以下のように解釈できる．まず，f_h を増加させることは基底ベクトルを保持する枠組みを増やすことに相当するが，f_h が少数の場合は出力画像に有効に寄与することが可能な基底ベクトルが生成されるため性能向上にも大きく貢献できる．しかし，f_h が大きくなるにつれて大きく寄与できる基底ベクトルは減少し，性能向上のカーブも緩やかになる．そして最後には，性能向上に寄与できる基底ベクトルも作れなくなり，復元性能としては頭打ちになる．これが図 4.3.4 及び，図 4.3.7 に対する解釈である．

上記解釈は SR-PDNN の更なるコストパフォーマンスを追求する上でも幾つかの可能性も示している．1 つは，学習されたネットワーク規模の妥当性評価である．例えば，$f_h = 800$ の場合，実は $f_h = 600$ でも同等の性能を出すことが出来たが，どの程度のネットワーク規模まで縮小できるかは図 4.3.7 をみれば推測することが出来る．これは，学習時間に非常に多くの時間を有する Deep Learning の分野において，魅力的な特徴である．しかも，SR-PDNN の場合，行列の積和演算によって各基底ベクトルに寄与する成分を特定

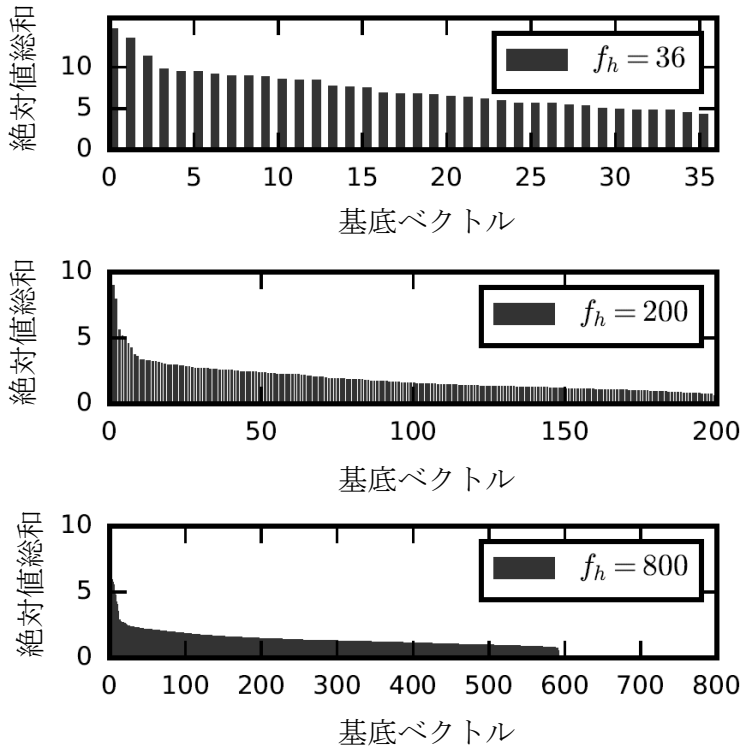


図 4.3.7： 基底ベクトルの絶対値総和

することが出来る．つまり，有効な基底ベクトルのみを抽出した後，再学習なしで対応する部分のみを切り出し，ネットワークを再構築できる．これは SR-PDNN がパッチベースの全結合型ニューラルネットによって構成されているが故に可能となる手法である．

なお，上記の考えは，コストパフォーマンスの更なる追求にも応用できる．例えば，基底ベクトルにおける絶対値の総和の大きさは出力画像に対する寄与にも影響することから，寄与の小さい基底ベクトルは敢えて削除することで更なるコストパフォーマンスの追求ができる可能性がある．また，性能は劣化するが評価関数に正則化項として基底ベクトルの絶対値の総和がなるべく小さくなるように指定ができれば，有効な基底ベクトル数としてはより減少する方向に作用し，更なるコストパフォーマンスを追及できる余地がある．

以上の通り，SR-PDNN は既に CNN ベースの手法である SRCNN 等と比較しても十分なコストパフォーマンスの増加が確認されているが，上記検討をさらに進めることで将来的にはさらにコストパフォーマンス性能を向上できる可能性がある．

4.3.3　学習データと復元性能の関係

前節では，SR-PDNN のハイパーパラメータである基底ベクトル数 f_h を変化させることでネットワーク規模に応じた復元性能の変化を検証した．本節では，SR-PDNN のネットワークは固定した状態で，学習データを変化させた場合における復元性能の変化を観察し，SR-PDNN を構成するネットワークアーキテクチャがどの程度までの復元性能を発揮できるのかといった部分について考察する．

表 4.3.7 は本実験で使用した実験条件である．通常，ニューラルネットを用いた学習と検証では学習データと検証データを分けて行い，学習データによって最適化されたネットワークの汎用性を検証データで評価するというのが一般的である．ここではネットワークが持つ復元能力のポテンシャルと汎用性を評価するために，敢えて学習データに検証データを用い，最も学習効果が高い正解データによる学習時の性能を確認する．具体的な実験条件は，拡大率 2 の条件において，図 4.3.8 にある butterfly 画像を復元対象と見做した状態で，学習データに butterfly 画像のみを用いた場合と，butterfly 画像を含む Set5 を用いた場合，そして，butterfly 画像を含まない 91 images で学習した結果を比較した．

表 4.3.7：実験条件

項目	拡大率: 2	拡大率: 2	拡大率: 2
入出力パッチサイズ	6 × 6	6 × 6	6 × 6
特徴次元数 f_l	200	200	200
基底ベクトル数 f_h	800	800	800
学習用データセット	***butterfly***	***Set5***	**91 *images***
学習用データのパッチ数	**60,516**	**133,879**	**4,916,516**
学習データ生成時のステップ幅	2	2	2
復元処理時のステップ幅	2	2	2

図 4.3.8：Set5

表 4.3.8：Set 5 復元性能比較

Set5 Scale: x2	Bicubic	ScSR	SRCNN (9-1-5)	SRCNN (9-5-5)	VDSR	SR-PDNN (butterfly)	SR-PDNN (Set5)	SR-PDNN (91images)
baby	37.07	38.45	38.30	38.54	38.72	37.81	38.76	38.53
bird	36.81	40.56	40.64	40.91	42.44	38.58	41.55	41.38
butterfly	27.43	31.31	32.20	32.75	34.45	36.08	34.74	33.48
head	34.86	35.73	35.64	35.72	35.91	34.95	35.8	35.77
woman	32.14	34.95	34.94	35.36	36.06	34.26	37.07	35.64
average	33.66	36.20	36.34	36.66	37.51	36.34	37.58	36.96

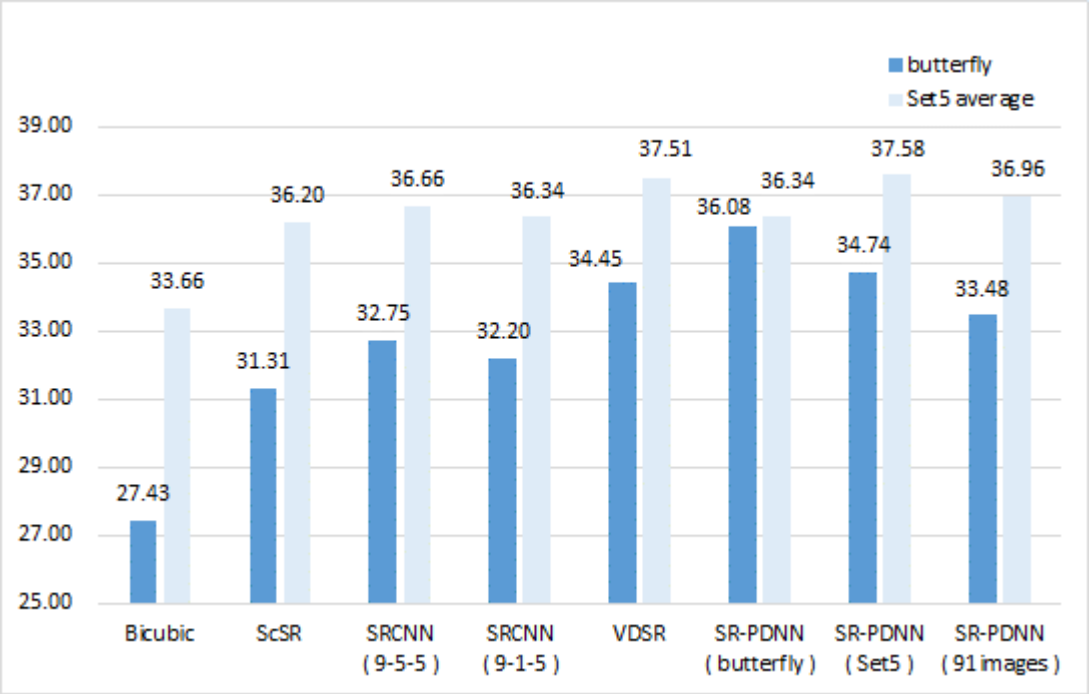


図 4.3.9: butterfly と Set5 の復元性能比較

表 4.3.8 は Set5 の復元性能に関する実験結果と各手法との比較を示している．また，図 4.3.9 は表 4.3.8 から注目画像である butterfly と Set5 の平均 PSNR を抜き出して図示したものである．

これらの実験結果から，まず，butterfly のみで学習した場合は 36.08 dB まで復元できることが分かる．この数値は VDSR と比較しても高い数値であり，SR-PDNN のアーキテクチャは butterfly 画像に関して最大でこの程度までの復元が出来るということを意味している．これは逆にいえば，91 images などの一般の学習用画像セットで学習したとしてもこの数値を超えることは先ずなく，ネットワークが持つ最大性能がどの程度かを知る上で重要な実験となる．なお，butterfly に特化されたネットワークが他画像で全く効果が無いわけでもなく，表 4.3.8 を見ても分かるように butterfly 以外の画像もそれなりに（偶然にも SRCNN 9-1-5 相当には）復元されており，数値としては入力である Bicubic をい

ずれも上回っている．これは，SR-PDNN の学習手段が超解像という課題に対して主として汎用的な成分に対して学習が行われていることを意味している．一方，Set5 の復元性能が最も高くなる場合はいずれかとなると，やはり，Set5 を学習対象として選んだ場合である 37.58 dB が最も高くなる．しかし，Set5 で学習した場合の butterfly は当然ながら butterfly のみで学習した場合と比べて数値は低下している. その傾向は 91 images で学習した場合も同様であり，Set5, butterfly による学習と比較しても共に数値が低下している.

これらの実験結果は限られたリソースでどこまで汎用性を持った学習が出来るかという点を見極める上で重要である．例えば，VDSR の性能を見ると，SR-PDNN を少ないデータセットである Set5 のみで学習した結果よりやや劣るが，さらに学習データの多い 91 images で学習した結果よりも上回っている．これは，VDSR が SR-PDNN よりも非常に大規模なネットワークを有しているため，汎用化による性能劣化をあまり起こさずに性能を維持できていると解釈することができる．一方で，SR-PDNN も用途を絞れば VDSR 相当の性能を出すことができることもわかることから，結局のところ，ニューラルネットで実現したい想定パターンがどの程度存在するかによってネットワーク規模は適切に選ぶべきであり，そのパターンを表しきれない場合は専門性と汎用性のトレードオフによって性能が定まってくるといえる.

上記の考察は，超解像を実際に応用する場合には，用途にあった画像で学習することが性能向上の面でも, ネットワーク規模削減の面でも重要であるという結論となる. 例えば，応用シーンとして複数のケースが存在する場合には，単純にネットワーク規模を増やすよりもそれぞれのシーンに特化したパラメータを算出し，適用シーンに応じてパラメータを入れ替えるという手法も考えられる．これは様々なモダリティ画像が表示される医療モニターには想定されうるケースである．また，ネットワークパラメータをメモリ上に展開すれば，これらのパラメータ切り替えもポインタ切り替えによって行えるため，ハードウェアコストを増加させずにより幅の広いケースに対応させられる可能性がある．これらの具体的なシステムアーキテクチャの設計もまた今後 SR-PDNN の実装を検討していく上での課題となる.

4.4　結言

本章では，本論文の提案手法である SR-PDNN について，その有効性検証と性能解析を行い，今後の課題について述べた．結論として，SR-PDNN は従来の CNN ベース手法に比べ非常に高いコストパフォーマンスを実現できていることが確認された．これは，SR-PDNN が超解像という問題に対してよりシンプルな状況で最適化学習できた結果であると捉えることができる．また，CNN にはなかった特徴として，学習後の基底ベクトルを観察することで，対象学習画像に対するネットワーク規模の十分性について見定めることが出来る点，そしてさらなるコストパフォーマンスの追求が出来る点についても議論した．本章で述べた内容をまとめると，以下の通りとなる．

4.2 SR-PDNN の性能評価では，まず，復元性能の評価では 91 images を学習データとした同一学習条件での比較結果から，SR-PDNN が高い復元性能を持つことを述べた．またコストパフォーマンスの比較では大規模ネットワークである VDSR を含む CNN ベース手法に対して，復元性能とハードウェア実装コストの両面からコストパフォーマンスを評価し，SR-PDNN が優れたコストパフォーマンスを発揮できることを示した．

4.3 学習条件に応じた性能解析では，まず，補間画素位置を特定することによる性能変化を学習データ生成時のステップ幅と復元処理時のステップ幅を変化させることで見極めた．結果，補間画素位置の特定はピーク性能の向上には寄与するものの対象画像位置のロバスト性という面では低下することが確認された．また，SR-PDNN のネットワーク規模を決定するハイパーパラメータの一つである基底ベクトル数 f_h を可変させた場合，ネットワーク規模の増加に伴い復元性能も向上することを示した一方で，性能がある一定の水準に達すると頭打ちになることも示した．ここで，頭打ちとなる原因は基底ベクトルの絶対値の総和が 0 以上となる有効基底ベクトルの総数を観察することで理解することが出来，拡大率が 2 の場合は $f_h = 600$ 以上に設定しても有効ベクトル数に増加がしないことを原因として示す一方で，学習後の有効基底ベクトル数の評価と再構築からネットワーク規模の妥当性と更なるコストパフォーマンスの追求が可能である点を示した．さらには，学習データと復元性能の関係として，学習データを特化させることで特定状況によってはさらなる性能向上が見込めることや，専門性と汎用性のトレードオフはネットワーク規模によっても変化することを示し，実応用面における更なる検討課題についても言及した．

参考文献

[4-1] C. Dong, C.C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” European Conference on Computer Vision, pp.184–199, Springer, 2014.

[4-2] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution as sparse representation of raw image patches,” Computer Vision and Pattern Recognition, 2008, CVPR 2008, IEEE Conference on, pp.1–8, IEEE, 2008.

[4-3] M. Bevilacqua, A. Roumy, C. Guillemot, M.L.A. Morel. “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” BMVC, 2012

[4-4] Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Curves and Surfaces, pp. 711{730. Springer (2012)

[4-5] J. Johnson, A. Alahi, ,“Perceptual Losses for Real-Time Style Transfer and Super-Resolution,” CoPR ,2016.

[4-6] J. Kim, J.K. Lee, and K.M. Lee, “Accurate image super-resolution using very deep convolutional networks,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1646–1654, 2016.

[4-7] D. Martin, C. Fowlkes, D. Tal, and J. Malik. “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” ICCV, 2001.

[4-8] Kodak image data set: http://www.cs.albany.edu/~xypan/research/snr/Kodak.html

第 5 章 結論

5.1 各章の総括

第 1 章 序論 では，映像の高解像度化への社会的期待が高まった背景と，リアルタイム処理の必要性について述べた．また，表示機器としての「超解像」とその技術動向の概略を述べ，その上で近年主流となってきた Deep Learning を用いた高性能な超解像システムを構築する上での課題について概要を述べた．さらに，本研究での目指す超解像システムの概要とその検討方針についても概要を示した．

第 2 章 低遅延リアルタイム処理の課題と関連研究 では，本研究で目指す超解像が持つべき特性である低遅延リアルタイム処理について，その意味を具体的に説明し，映像表示システムの開発という観点ではハードウェア実装による実現を目指すことが現実的であることを説明した．その上で，リアルタイム映像システムの構築という観点から従来の超解像技術を俯瞰し，当該研究領域における従来手法の課題を示した．具体的には，CNN ベース手法が持つコストという課題をパッチベース手法で克服すること，End-to-End を追求したシンプルな学習環境を実現できるアーキテクチャを考案することで機械学習の恩恵を最大限引き出すことを取り組むべき課題として定めた．

第 3 章 パッチベース型 DNN による超解像システム では，本研究の課題である低遅延リアルタイムの実現に向け，提案手法である SR-PDNN の設計思想を記述した．SR-PDNN はパッチ画像を入出力にもつ PDNN を核に構成されており，PDNN は画像から特徴を抽出する入力層，特徴量を変換する推定層，出力を生成する復元層から構成されることを述べ，いずれの階層も超解像という問題によりシンプルに取り組むことを目指し，極力不要なパラメータは除外した．また，学習データの生成や復元時のパッチ選択においても補間画素位置を考慮したステップ幅を設定することで不要なばらつきを取り除き，超解像問題をよりシンプルにすることを考慮したシステムを考案した．

第 4 章 シミュレーションによる性能評価と解析 では 3 章で提案した SR-PDNN について，その有効性検証と性能解析を行い，今後の課題と展望についても述べた．結論として，SR-PDNN は従来の CNN ベース手法に比べ非常に高いコストパフォーマンスを実現できた．これは超解像という問題に対してよりシンプルな最適化学習が行えた結果であるとみなせる．また異なる学習条件下における SR-PDNN の復元性の変化や他手法との復元性能の比較から，学習データ，ネットワーク規模と復元性能の関係性を明らかにし，さらなるコストパフォーマンスの可能性と今後の展望や課題についても述べた．

第 5 章：結論 では，各章の主旨を統括し，低遅延かつリアルタイム処理に適したニューラルネット型超解像システムの研究に関する本論文の結論を述べた．

謝　辞

本論文は著者がEIZO株式会社に在籍しながら，金沢大学大学院 自然科学研究科 電子情報科学専攻にて行った3年間の研究成果をまとめたものです．

まず，主任指導教員である 金沢大学 理工研究域電子情報通信学系 今村幸祐 准教授には本研究の遂行から本論文の作成に至るまで，終始適切な助言とご指導を賜りました．ここに深く感謝の意を表します．

また，博士号取得を目指すきっかけを与えくれた 金沢大学 理工研究域電子情報通信学系 松田吉雄 教授と，金沢大学 理工研究域フロンティア工学系 平野晃宏 講師には，本研究の遂行にあたり数々の貴重なご指導とご鞭撻を賜りました．深く感謝いたします．

また，金沢大学 電子情報通信学系 北川章夫 教授，秋田純一 教授，南保英孝 准教授には，お忙しい中，審査をお引き受け頂きました．厚く御礼申し上げます．

また，EIZO株式会社には就学に必要な費用の一部を援助して頂きました．特に，EIZO株式会社 村井雄一 専務，橋本雅之 専務，堅田秀雄 部長 には，業務の傍ら進学することを許可して頂き，博士号取得を目指すことが出来ました．深く感謝いたします．

最後に，まだ子供が小さな家庭環境で社会人学生として研究に励むことに理解を示し，日々支えてくれた妻 青木千絵と，共に過ごす時間が限られ寂しい思いをさせてしまった長女 紗英，次女 更紗，長男 陽に心より感謝いたします．

令和元年9月

青木 玲央