

U-net を用いた異常検知による肺炎の検知

長村 徹¹ 徳永 旭将¹ 大北 剛¹

概要: 本研究では U-net[1] を利用した異常検出によって, 胸部レントゲン画像から肺炎の検出を行う手法を検討する. U-net を利用した異常検出では, 学習していない画像が入力された場合に, 入出力画像の差異が大きくなることを利用して, 異常画像を検出する. 肺炎を検出するために考えられるのは, 正常画像と肺炎画像の 2 クラス分類だが, 本研究ではさらに踏み込んで, 肺炎画像内に含まれる陰影という肺炎の特徴を取り出すことを目的とする. この陰影を U-net で検出するための工夫として, 元画像から小領域を取り出したパッチを用いて, U-net を訓練・テストする. 訓練時には正常画像の肺内部のパッチの復元だけを学習した U-net 作成し, テスト時には肺炎画像のパッチを入力する. 小領域を切り出したことでパッチに含まれる情報は限定され, その情報の違いによりテスト時には U-net の入出力画像の差異に違いが生まれる. この違いを利用して, 肺炎画像のどこに正常画像にはない陰影という異常が現れるのかを検知する.

Pneumonia Detection Using Anomaly Detection by U-net

TORU NAGAMURA¹ TERUMASA TOKUNAGA¹ TSUYOSHI OKITA¹

1. はじめに

近年では増え続ける医用画像の診断を自動化するためのソフトウェアなどの開発が盛んに行われている [7] [8]. このためのアプローチとして, 畳み込み層ニューラルネットワーク (Convolutional Neural Network: CNN) [4] によるクラス分類や異常検知などの機械学習手法がある. クラス分類では各クラスに対応する大量の画像を用意して CNN の学習を行う. この場合, 出力値はどのクラスに分類されるかという確率値である. 一方で異常検出ではあるクラスに対応する画像のみを学習に用い, テスト時にそれ以外の画像が入力された際に異常だと判断する. 異常検出の出力値は画像であり, 入力画像と出力画像の差異によって異常な画像を判断する.

一方でクラス分類や異常検知ができて, これらの技術では対象の疾患の特徴がどこに表れているかを判断することはできない. 例えば, 胸部レントゲン画像を用いて肺炎の診断を行う際に医師は, 肺炎に罹患した肺に表れる正常画像にはない陰影という特徴を見つけだしこれを指標とする. だが上記のような機械学習手法を適用した場合は, 何に目してクラス分類や異常検知を行っているのか不明で

ある. 症状がある場所やその状態を判断することは, 治療法を決定するうえで重要な情報となるはずである.

そこで本研究では胸部レントゲン画像から肺炎の特徴である陰影の場所を検出することを目的とする. このために U-net[1] による異常検知を, 元画像からある小領域を切り取ったパッチ画像で行うというアプローチをとる. パッチ画像を用いることで画像内に含まれる情報が限定されるため, これを利用して異常部分を検出することができる. ここで用いる U-net は医用画像のセマンティックセグメンテーションのために開発された手法であるが, 画像内の被写体の位置情報を入出力間で保存できるという, 異常検知にも有効な特徴があるためこれを用いる.

2. 手法

2.1 U-net

U-net[1] は画像のセマンティックセグメンテーションのために開発された畳み込みニューラルネットワークの一種である. セマンティックセグメンテーションとは画像を画素単位でクラス分類するタスクである. U-net は細胞などの医用画像のセマンティックセグメンテーションで成果を上げた手法である.

U-net は画像のエンコード (符号化) とデコード (復号

¹ 九州工業大学

化)を行うアーキテクチャを持つ。エンコード部分では畳み込み層とプーリング層によって画像を符号化する。デコード部分では畳み込み層とアップサンプリング層によって画像を復号化する。U-netはこの過程で入力画像の特徴抽出を行い、セグメンテーション結果を出力する。

セマンティックセグメンテーションでよく使われるFCN (Fully Convolutional Network) [2]と比較してU-netで特徴敵なのは、エンコード部分の畳み込み層が学習した重みをデコード部分にコピーするスキップ結合があることである。スキップ結合は畳み込みやプーリングで失われてしまう、被写体の位置情報を補完する効果がある。U-netではスキップ結合でセマンティックセグメンテーションにおいて重要な被写体の位置情報を保持しつつ、畳み込み層により特徴を抽出することを可能にしている。

2.2 オートエンコーダを用いた異常検出

オートエンコーダ [3]とは入力画像をそのまま生成することを目標としたニューラルネットワークの一種である。U-netと同様に画像を符号化して復号化するエンコーダ・デコーダ構造を持つ。一般的にオートエンコーダで着目するのは、画像をエンコードした結果得られる、元画像よりも低次元の情報である。

オートエンコーダでは入力画像をそのまま再現するための学習がされているため、これを利用して異常な画像を見分けることが可能となる [9] [10]。学習がうまくいったオートエンコーダは、学習時に入力として与えた正常画像でテストをすれば、高い精度で画像を復元できる。だが、学習時には入力として与えたことのない画像 (異常画像) でテストをすると、復元された画像の再現率が低くなる。この性質を利用して、再現率が下がった画像を異常な画像だと識別することで異常検出を行う。なおオートエンコーダによって復元された画像を生成画像と呼ぶこととする。

3. 肺部分異常検出のアルゴリズム

3.1 実験に用いるデータセット

Kaggleより取得した、Chest X-Ray Images (Pneumonia) dataset *1 を用いて実験を行う。胸部レントゲン画像から、正常な肺の画像と肺炎に罹患した肺の画像を見分けるというコンペティションのためのデータである。提供されている画像のサイズはまちまちで扱いづらいので 256×256 に縮小して使用する。図1に画像例を示す。

事前実験としてこのデータセットを対象として、正常画像と肺炎画像を2クラス分類することに取り組んだ。訓練データとして正常画像を1341枚、肺炎画像を3875枚準備した。画像分類のためのアーキテクチャとして3層の畳み込み層を持つCNNを用いた。カーネルサイズは

3×3 、各畳み込み層のあとに maxpooling 層を設け、最後の maxpooling 層の後にバッチ正規化 [12] をした。その後に FC 層が1層続き、ドロップアウト [11] 率が0.5のドロップアウト層の後、再び全結合層を用いた。損失関数は binary cross entropy を使い、Adam [5] を使った最適化を行った。最適化の際に不均衡データに対処するため、正常画像に対する学習率を肺炎画像に比べて大きくする処理を加えている。これは keras を用いて python で実装した。その結果正解率は87.6%となった。この cnn に grad cam [6] を適用した結果を2に示す。青色に着色されている部分がCNNが注目している場所にあたるが、肺内部には注目できていないことがわかる。そこで本研究ではパッチを用いて、訓練画像から骨などの無駄な情報は除去する。

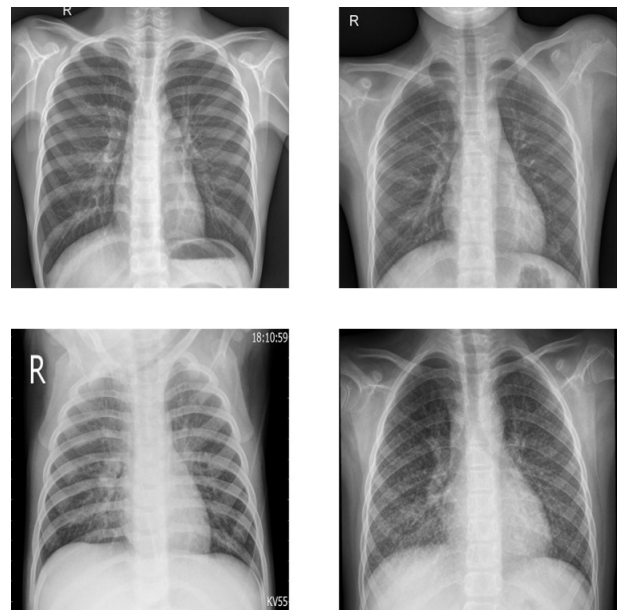


図1 画像例：上段が正常・下段が肺炎に当たる画像

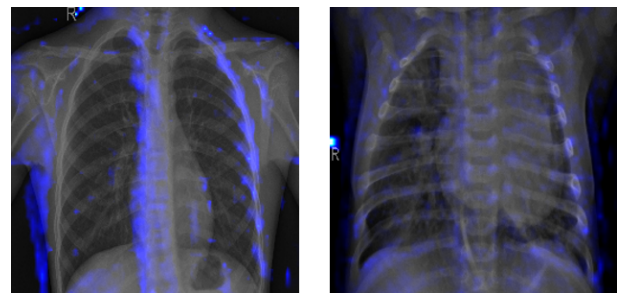


図2 2クラス分類をするCNNに grad cam を適用した結果

3.2 画像の類似度：SAD

オートエンコーダの性能評価の指標として入力画像と生成画像の SAD を用いる。参照画像を $I(x, y)$ 、テンプレート画像を $T(x, y)$ とすると SAD は

$$SAD = |I(x, y) - T(x, y)| \quad (1)$$

*1 <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>

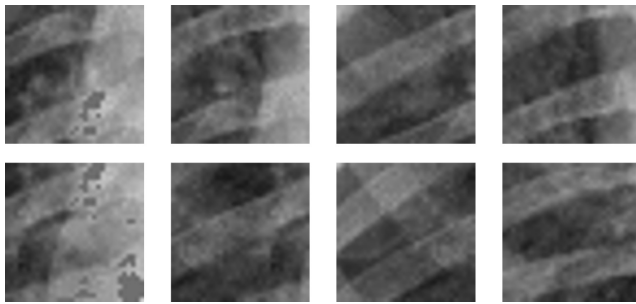


図 3 正常パッチの例



図 4 肺炎パッチの例

となる．例えば SAD が小さいほど 2 つ画像の差異は小さいことになる．よって入力画像と生成画像の SAD が小さいほどオートエンコーダの性能が高いということになる．

3.3 提案するアルゴリズム

肺炎の特徴である陰影の場所を検出するためのアルゴリズムを示す．まず対象のデータセットから正常画像と肺炎画像それぞれで特徴の異なるパッチ画像を作成する．正常画像では肺のみをセマンティックセグメンテーション²で取り出すことで、肺内部のみのパッチを作成できる．一方、肺炎画像には陰影があることが原因となり、このセマンティックセグメンテーションの手法は適用できない．よって肺内部のみのパッチは作成できないので、元画像から一定の大きさでパッチを全探索的に取り出す．すると肺炎画像のパッチは以下のような 3 種類の部分を元画像より切り出すことになる．

- 肺内部で健康な部分を切り出したパッチ
- 肺内部で肺炎の症状である陰影が現れた部分を切り出したパッチ
- 肺の外の内臓や背景を切り出したパッチ

本研究では U-net をオートエンコーダとして用いることで異常検出を行う．以下にそのための訓練とテストの手順を示す．

- (1) 正常画像のパッチで U-net を訓練する．
- (2) 肺炎画像のパッチを U-net に入力して、その出力パッチを得る．
- (3) 入力パッチと出力パッチの画像の SAD を計算する．
- (4) SAD を横軸に度数ヒストグラムを作成し、肺炎画像のパッチの種類（健康な肺・肺炎の肺・骨や背景）に対応する閾値を決定する．

- (5) 閾値によって 3 種類のパッチを見分けることができる．

ここで画像の SAD によって、肺炎画像の 3 種類のパッチを識別する方法論を説明する．U-net が正常な肺の内部しか学習していないことから、肺炎のパッチを入力するとこれを再現できず、入出力画像間で SAD が増加する．しかも肺炎パッチは異なる 3 種類の特徴を持つので、入出力画像間の SAD が 3 種類それぞれで異なる．よって表 1 に示すような画像の SAD の違いを利用して、閾値を設ける

ことで 3 種類の肺炎パッチを識別できる．

例えば、肺炎のパッチが肺の内部のみしか含まない場合には、学習した正常画像のパッチと全く同様の画像を入力していることになるため、SAD は全体と比較すると低くなるはずである．また肺炎のパッチが肺内部のみしか含まない場合にも、パッチで切り出した部分が肺炎の特徴である陰影を含んでいる場合は、陰影を含んでいない、つまり健康な部分を切り出した場合に比べると SAD は高くなる．一方で肺炎のパッチが骨や内臓・背景しか含んでいない場合は、学習した正常画像のパッチとは全く異なるものを入力しているため、SAD は平均と比較すると高くなるはずである．このように SAD の値によって閾値を設けることによって、入力したパッチ画像に移っているものを分類することができるはずである．以上の方法論を表 1 にまとめる．ただこれらを複数含んだ画像もあることが問題となり、閾値を設定するのは難しい．

肺炎パッチの切り出したもの	入出力画像の SAD
健康な肺内部のみ	学習した画像と同程度
肺炎に罹患した肺内部のみ	上より大きく、下より小さい
骨や内臓、背景など肺以外の部分	上 2 つより大きい

表 1 SAD の大きさによる画像識別の方法論

3.4 正常画像のパッチ作成

元の画像には肺の周りに骨や内臓などの肺炎には関係のない情報が多いので、肺の内部のみの特徴を学習したい．これらをそぎ落とした情報をオートエンコーダに与えることで、肺の中に存在する肺炎の陰影だけを取り出すことを狙う．このためのアプローチとして元画像から肺内部だけを取り出したパッチを作成することを考える．なお本研究においてパッチとは、元画像から指定したサイズの小領域を切り出して画像化したものを指すこととする．

正常画像のパッチの作成手順を説明する．まず元画像から肺部分だけを抽出し、そのうえでパッチが完全に肺の内部しか含まないような画像を作成する．元画像から肺部分だけを取り出す方法として、学習済みの U-net ^{*2} を用い

^{*2} <https://github.com/imlab-iiip/lung-segmentation-2d>

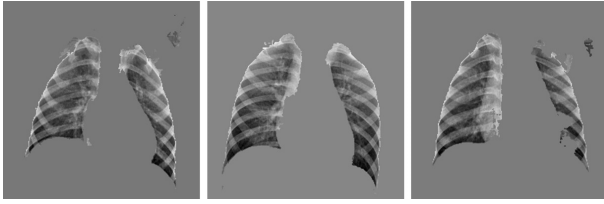


図 5 セグメンテーション画像の例

る。これは図 5 のような画像を出力する。肺内部の画像は前景として画像に残り、残りの部分は背景として同じ色で塗りつぶされる。以後この画像をセグメンテーション画像と呼ぶこととする。

セグメンテーション画像の背景が同じ色で塗りつぶされていることを利用して、肺内部のみを取り出したパッチを作成する。背景はすべて同じ色なので、この色の画素がパッチに全く入らないように元画像から切り出す。この手順により、元画像から肺内部のみを含んだパッチが作成される。作成したパッチの画像例を図 3 に示す。

3.5 肺炎画像のパッチ作成

肺炎画像では 2.2 の正常画像のパッチ作成の手順と同様にはできない。なぜならば肺部分を抽出する学習済みの U-net は、肺炎画像には対応しておらず肺のみを取り出すことはできない。よって肺炎画像のパッチは 32×32 の大きさの画像を元画像から全探索的に取り出す。このようにして取り出したパッチには以下の 3 種類の情報が含まれる。

- 肺内部
- 骨や内臓
- 背景

切り出す位置によって上の情報のうち 1 つしか含まない、もしくは複数含むパッチができる。作成した肺炎画像のパッチ例を図 4 に示す。

3.6 異常検知を行う U-net のアーキテクチャ

U-net 構造は一般的にはセグメンテーションのために開発されたものだが、これをオートエンコーダとして使う。オートエンコーダに比べて特徴的なのは、エンコーダの重みをデコーダにコピーするスキップ結合があることである。これにより入力画像の位置に関する情報を欠損せずに生成画像を作成することが期待できる点である。

画像の復元のために適切な損失関数を設定する必要がある。本実験では集合同士の類似度を測る指標である Dice 係数を利用した、Dice 損失関数を設定した。ある集合 A と別の集合 B を考えると、Dice 損失関数は以下のようになる。

$$Dice_Loss(A, B) = -\ln \frac{2|A * B|}{|A| + |B|} \quad (2)$$

Dice 損失関数を画像データに適用する。 $I(x, y)$ を画像 I のピクセル (x, y) での輝度値とする。また $J(x, y)$ を画像

J のピクセル (x, y) での輝度値とする。

$$Dice_Loss(I(x, y), J(x, y)) = -\ln \frac{2|\sum_x \sum_y I(x, y) * J(x, y)|}{\sum_x \sum_y (|I(x, y)| + |J(x, y)|)} \quad (3)$$

図 6 に使用した U-net の構造を示す。U-net の前半部分は画像をエンコード (符号化) し、後半部分は画像をデコード (復号化) する。いずれも畳み込み層のフィルタサイズは 3×3 で、画像サイズが入出力で同じになるようにゼロパディングをする。チャンネル数はプーリング層を通るたびに倍になり、アップサンプリング層を通るたびに半分になる。ここでプーリング層とはある小領域に対して最大の画素値を選択して出力することで画像サイズを半分にする処理である。一方でアップサンプリング層とは、ある一つの画素を小領域に複製することで画像サイズを倍にする処理である。

入力画像のサイズは 32×32 である。エンコード部分でプーリングを繰り返すたびに画像のサイズは半分になり、最終的には画像サイズは 8×8 となる。一方でデコード部分ではアップサンプリングを繰り返すたびに画像サイズが倍になるため、最終的な出力画像のサイズはもとに戻り 32×32 となる。エンコード部分は各畳み込み層の後に maxpooling 層、デコード部分では upsampling 層を設けている。この U-net を Adam[5] を用いて最適化した。また keras を用いて python で実装した。

3.7 異常検出を行う U-net の訓練とテストの手順

肺炎のパッチを異常として検出するために、U-net の訓練時には正常のパッチのみを学習させる。つまり訓練時の入出力画像はともに正常のパッチで、正常のパッチを入力すると正常のパッチが出力される U-net ができる。一方テスト時には、この正常のパッチのみを学習した U-net に肺炎のパッチを入力する。U-net は正常のパッチしか学習していないので、肺炎のパッチを入力した場合にはその特徴を再現できず、入力画像とは異なる画像が出力されるはずである。次節以降で紹介するが本研究ではこれを利用した異常部分の検出を行う。

3.8 訓練データやテストデータとなるパッチ画像の作成方法や枚数

訓練データとなる正常画像のパッチを実際に作成すると、元の正常画像一枚につき平均 6 枚のパッチ画像が取れる。この時パッチ画像を切り出すためのウィンドウを動かす幅は、本実験では 20 画素ごとに設定する。データセットから元の正常画像となる画像を 100 枚抽出してパッチを作成すると、合計で 6050 枚の正常画像のパッチが作成できた。このパッチを、訓練データ:5750 枚・バリデーションデータ:100 枚・テストデータ 200 枚に分ける。

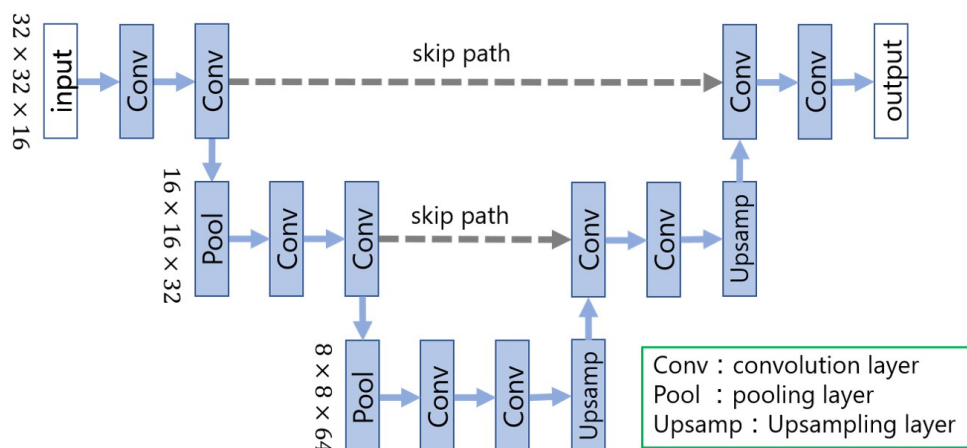


図 6 U-net の構造

異常検出の対象となる肺炎画像のパッチは、元となる肺炎画像を全探索して作成する。このとき正常画像に正常画像と同様に、ウィンドウを動かす幅を 20 画素ごとに設定する。32 × 32 の画像切り出しを 20 画素ごとに行うことで、元画像 1 枚から切り出せるパッチの枚数は 420 枚となる。元画像を 5 枚用意してパッチを切り出し、肺炎画像のパッチを 2100 枚用意した。

4. 結果

4.1 オートエンコーダに正常画像と肺炎画像のパッチをそれぞれ入力した場合の結果

訓練したオートエンコーダに正常画像のパッチのテスト画像を入力した結果を図 7 に示す。また訓練したオートエンコーダに肺炎画像のパッチのテスト画像を入力した結果を図 8 に示す。図は 1 行目が入力画像、2 行目が生成画像、3 行目が入力画像のヒストグラム、4 行目が生成画像のヒストグラム、5 行目が入力画像と生成画像の SAD となっている。ヒストグラムの横軸は画素値 (0~255) で縦軸は頻度である。

4.2 2100 枚の肺炎画像のパッチを対象とした結果

2100 枚の肺炎画像のパッチを訓練したオートエンコーダに入力して結果を解析した。2100 枚の入力画像と生成画像の SAD をとり、これを SAD の値ごとにヒストグラムにした。その結果図 9 のような結果が得られた。ヒストグラムの横軸は SAD の値で縦軸は頻度である。

4.3 パッチが元画像での位置を表示した結果

入力画像と生成画像の SAD を計算すると、SAD の値ごとにパッチがいくつかのグループに分かれるはずである。図 10 ではある範囲内の SAD を持つパッチが元画像のどこにあるのかを赤枠で表示している。

5. 考察

5.1 オートエンコーダの性能の検証

図 7 の 2 行目の生成画像をみるとオートエンコーダは入力画像をよく再現できているように見える。3, 4 行目のヒストグラムを見ても大体形が似通っているが、完璧に同じではないことが見て取れる。実際 SAD は 0 にはならない。値は 1 付近に集中しているものの、入力画像と生成画像で画素値に差があるということである。

図 8 を見るとわかるように、肺炎画像のパッチを入力した際の生成画像の復元が可能となってしまった。とくに 1 列目の画像については明らかに入力画像が背景であり学習時には与えていない画像であるので、オートエンコーダでは全く再現できないと考えていた。しかし生成画像やそのヒストグラムを見比べてみても明らかに画素値が異なる画像が生成されているわけではない。これは入力画像の解像度が 32 × 32 と低いので、オートエンコーダが再現しやすいことが理由としてあげられる。

5.2 オートエンコーダの生成画像と入力画像の SAD を異常検出や異常部分検出に用いることについて

肺炎画像のパッチに対する SAD は平均的に正常画像のパッチより高く 5 付近に集中していた。予想では肺炎画像のパッチの中でも肺内部の健康な部分を切り出しているものは、SAD が正常画像のパッチの SAD (1 付近に集中) に近づくと考えていた。肺炎のパッチを入力した際に SAD が予想通りの 1 に近い値となる画像はなかった。しかし SAD の平均値が異なることを利用して、パッチが正常画像と肺炎画像のどちらに由来するものか識別することは可能かもしれない。

実際に作成した 2100 枚の画像をオートエンコーダに入力し生成画像との SAD をとってみても、有効な分離境界面は見られないことが図 9 より分かる。このヒストグラム

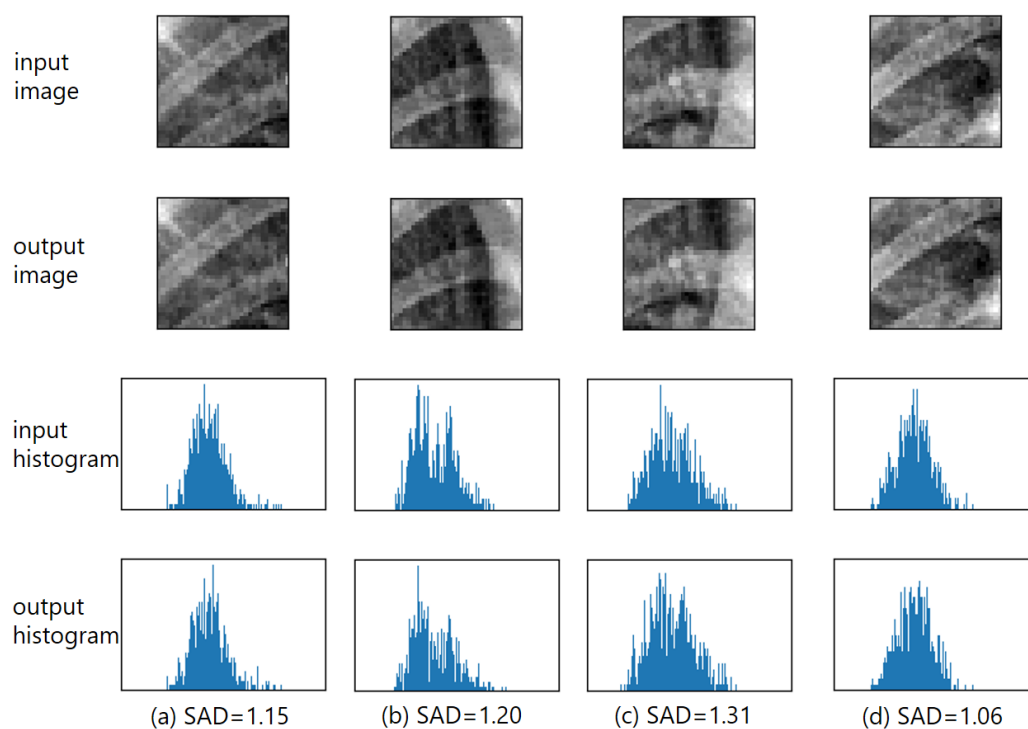


図 7 正常パッチに対するテスト結果

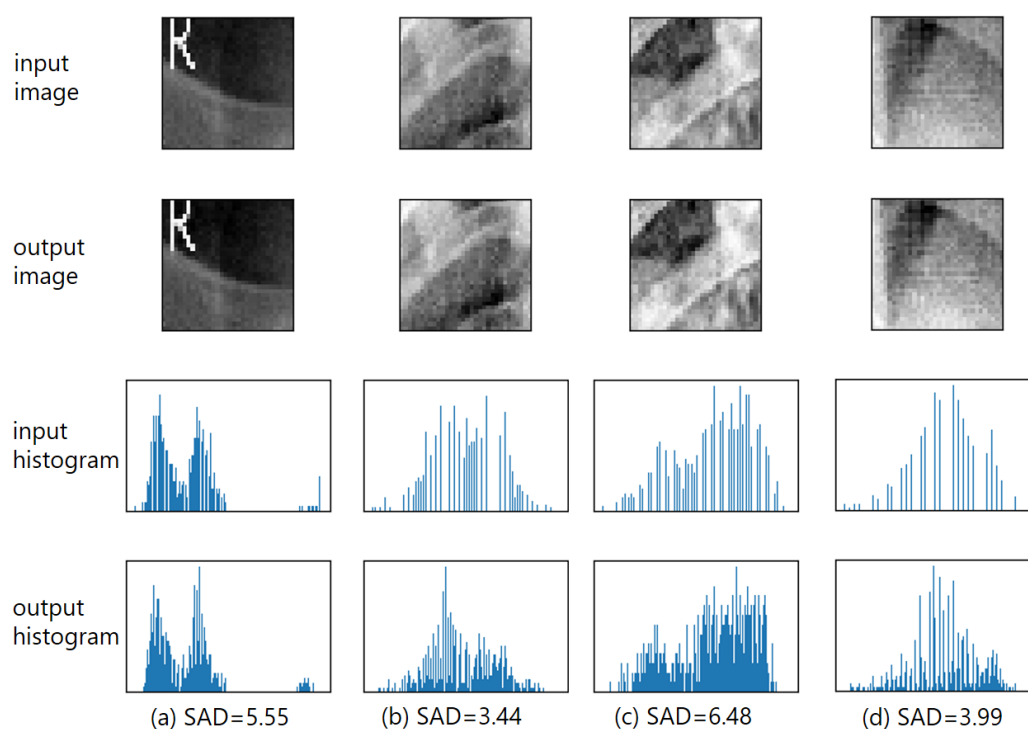


図 8 肺炎パッチに対するテスト結果

を作成すると表 1 で見たような 3 グループに対応するように、ヒストグラムに山が 3 つほど現れることを期待していた。しかし図 9 には山は一つしかなく、図 9 を参考にして異常部分検出のために SAD の閾値を定めることはできない。

図 10 では、SAD の範囲を適当に定めその範囲内に SAD が収まるパッチが元画像のどこから切り出されたものかを赤枠で示している。その結果 SAD をどのような値に設定しても赤枠のつく場所に規則性はないことがわかった。例えばある値の範囲では肺内部のみに赤枠がつくことを期待

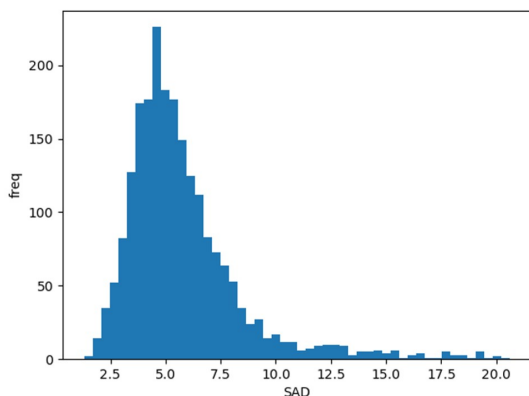


図 9 2100 枚の肺炎画像のパッチから得た SAD のヒストグラム

していたがそのような画像は見られなかった。

5.3 まとめと改善案

本実験ではオートエンコーダを用いた異常検出を、異常部分の検出に利用しようという取り組みについて紹介した。しかし SAD によって異常部分検出のための閾値を定めることはできなかった。一方で SAD の値によって少なくとも正常パッチと肺炎パッチの分類はできるかもしれないという結論に至った。

異常部分の検出ができなかった理由として最初にあげられるのが画像の解像度である。今回はデータが扱いやすいように低解像度の画像を用いたが、この時点で異常部分検出のために必要な情報が欠落していたと考えられる。

今後の方針としてパッチ画像が 3 クラスに分かれると予想していたことを利用してクラスタリングをすることも考えられる。例えば入力画像と生成画像の差分画像をクラスタリングすることで異常部分とそうでない部分のクラスタリングなどが可能かもしれない。また U-net 構造を持つオートエンコーダが提案手法に対して適切かはわからないので、他のネットワークアーキテクチャや学習手法を検討する必要もある。

参考文献

- [1] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, (2015).
- [2] Darrell, JLaESaT, J. Long, and E. Shelhamer. "Fully Convolutional Networks for Semantic Segmentation." IEEE T PATTERN ANAL 39.4 (2014).
- [3] Baldi, Pierre. "Autoencoders, unsupervised learning, and deep architectures." Proceedings of ICML workshop on unsupervised and transfer learning. (2012).
- [4] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. (2012).

- [5] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
- [6] Selvaraju, R. R., et al. "Grad-CAM: visual explanations from deep networks via gradient-based localization." arXiv website. arxiv.org/abs/1610.02391. Revised March 21 (2016).
- [7] Novikov, Alexey A., et al. "Fully convolutional architectures for multiclass segmentation in chest radiographs." IEEE transactions on medical imaging 37.8 (2018): 1865-1876.
- [8] Rajpurkar, Pranav, et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning." arXiv preprint arXiv:1711.05225 (2017).
- [9] Zhou, Chong, and Randy C. Paffenroth. "Anomaly detection with robust deep autoencoders." Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2017).
- [10] An, Jinwon, and Sungzoon Cho. "Variational autoencoder based anomaly detection using reconstruction probability." Special Lecture on IE 2.1 (2015).
- [11] Hinton, Geoffrey E., et al. "Improving neural networks by preventing co-adaptation of feature detectors." arXiv preprint arXiv:1207.0580 (2012).
- [12] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).

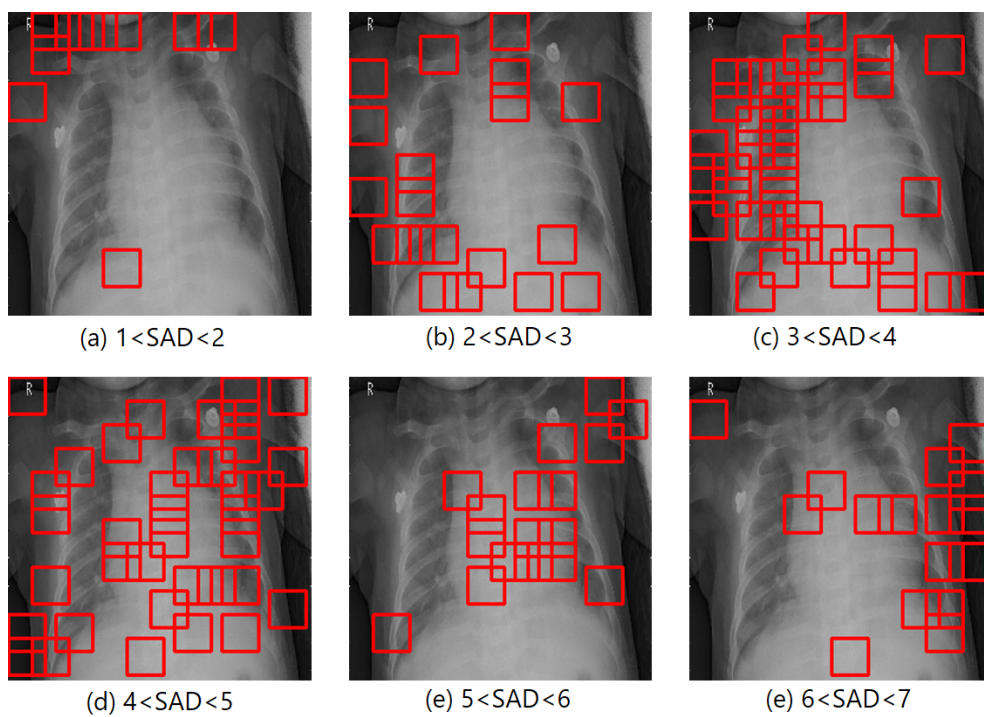


図 10 元画像でのパッチ画像の位置を SAD の値ごとに示した結果