Vol.2021-CVIM-225 No.43
2021/3/5

アニメ制作の品質管理工程における密な画素対応を用いた作画ミス検出

沖川翔太[1]　山口周悟[1]　森島繁生[2]

概要：アニメ制作の品質管理において，ミスの発見・修正を行うために膨大な量のアニメーション画像を精査しなければならず，制作現場の大きな負担となっている．そこで，精査するべき画像の枚数を減らし，負担を軽減することを本研究の目的とする．本研究では，アニメーション画像の連続性を利用して，色の塗りミスの検出を行う．まず，ターゲット画像とその 1 フレーム前の画像同士で画素ごとの意味的な対応を取る．ターゲット画像に色の塗りミス箇所がある場合，双方向の対応関係が取得できない箇所が現れる．このような箇所が検出された画像を異常画像とする．本手法では，カット内に 1 フレームだけ色の塗りミスが発生する場合に，ミスを検出することを可能とした．

キーワード：画像処理，異常検知，画素対応

1. はじめに

アニメ制作過程において色の塗り間違いや，人の指を 6 本描いてしまうなどの構造上の間違いといった作画ミスがしばしば発生する．制作過程の各工程においてミスのチェックが行われているが，そのチェックに漏れがある可能性がある．品質管理の工程では，制作されたアニメーションの最終チェックを行う．ここでミスのチェック漏れが発生するとそのミスが放送されてしまうため，この段階でミスを残さず発見する必要がある．しかし品質管理におけるミスの検出は，キャラクターの設定画とアニメーション映像を目視で比較するという方法によって行われているため，ミスを見逃してしまう可能性がある．さらに，一本のアニメーション動画は大量のフレーム画像で構成されている(24 フレーム×60 秒×30 分≒約 4 万フレーム)ため，1 話分のミスを検出するのに 3 人がかりで 3〜5 時間費やされており，膨大な負担となっている．そこで，あらかじめミスがありそうなフレームを自動的に検出することによって，品質管理の負担を軽減することを本研究の目的とする．

異常検知を行う従来手法としては，Variational Auto-encoder (VAE) を用いる手法 [1] や Generative Adversarial Network (GAN) を用いる手法 [2] などが存在する．これらの手法は事前に大量の画像を必要する．これらの手法を本研究の対象に適用させる際に，事前に用意する画像はそれまでに制作されたアニメーション画像や設定画が挙げられるが，登場する頻度が少ないキャラクターは事前用意できる画像の枚数が少ない．また 1 話だけしか登場しないようなキャラクターの場合には，事前に用意できる画像が設定画しかないということが発生する．これらのようなケースにおいては事前に大量の画像が必要となる異常検知を適用することは出来ない．しかし，登場する頻度の少ないキャラクターにおける異常検知の方が実際の制作現場においては需要が高い．登場する頻度の高いキャラクターの品質管理を行う際は，チェックする人は大量にチェックした経験があるため，キャラクターの特徴を記憶することができることによりチェックのミスは少ない．一方，登場する頻度が少ないキャラクターの品質管理はそれまでにチェックを行うことが少ない，または全くないため，チェックする人がキャラクターの特徴を把握することが難しく，チェックのミスがより発生する．

そのため本研究では登場する頻度の少ないキャラクターの異常検知に対応できるように，事前に大量の画像を必要とせず，アニメーションの連続性を利用して連続するフレーム同士の画素対応を取って 1 フレームだけ色塗りのミスが発生する場合の異常検知を行う．

2. 関連研究

2.1 異常検知

異常検知を行う手法としては VAE や GAN を用いたものが存在する．Dehaene らの手法 [1] では，正常データのみを用いて VAE の学習を行う．これにより，ネットワークは正常値のみが出力されるように構築される．テスト時には，入力した異常データに最近傍の正常データを出力し，入力と出力の差分を取ることによって異常箇所を検出することが可能である．さらにエネルギー関数に正則化項を加えることで，より出力が入力に近い画像となるようにしている．Schlegl らによる AnoGAN [2] では，正常データのみを用いて GAN の学習を行う．テスト時には入力画像に対応する潜在変数を反復法で求め，その潜在変数から訓練済の GAN によって再構成された画像と元の画像を比較する．異常検

1 早稲田大学
Waseda University
2 早稲田大学理工学術院研究所
Waseda Reserch Institute for Science and Engineering

知は Discrimination Loss と Residual Loss を組み合わせて算出される Anomaly Score によって行われる．

これらの手法は事前に大量の画像が必要とされるが，品質管理においては大量の画像が必要となる異常検知の需要は低く少数の画像で行う異常検知の需要が高いため，本研究の問題設定に合わない．

2.2 画像認識

Krizhevsky らは，ディープラーニングを用いた画像認識手法 AlexNet [3] を提案した．AlexNet は畳み込み層を用いることで 2012 年の ImageNet コンペティションにおいて従来の画像処理手法による記録を大きく更新する高い性能を示した．その後，Simonyan らによって 3×3 の小さな畳み込みを利用した畳み込み層と全結合層をつなげて層を深くすることによって，より画像認識問題において高い性能を出すことができる VGG [4] が提案された．VGG は画像認識問題に限らず画素対応などの様々な分野に用いられている．Illustration2Vec [5] は画像認識問題の中でもイラストの特徴をベクトル表現することを目的としている．VGG11 のネットワークをベースとして全結合層を畳み込み層に置き換えることによって，イラストに含まれている特徴をネットワークが判断できるようにした．大量のイラストによって訓練されており，イラストに描かれているキャラクターが「目が青い」や「黒髪である」などの特徴がどの程度の確率で存在するかを出力することが可能である．

2.3 画素対応

画像間の対応を取る手法としては，SIFT Flow[6]や PatchMatch[7]などの方法が存在する．SIFT Flow においては，Scale-Invariant Feature Transform (SIFT) [8] 特徴量を用いて画像間の密な対応を取ることを目的としている．SIFT 特徴量を用いているため，照明変化，回転，拡大縮小に頑強であるという特徴がある．この手法は色の勾配に注目しているが，イラストは色の勾配のない場所が多く，SIFT 特徴量が記述子として適切ではない．PatchMatch では，7x7 程度のサイズのパッチを用いた画像間の密な対応付けを，高速に計算することが可能である．しかし，特にイラストのような画像内の勾配の少ない画像に対しては，大域的な対応関係を適切に取得することが困難である．

ディープラーニングを使用する方法としては，Liao らによる Deep-Image-Analogy [9] や Aberman らによる Neural Best-Buddies [10] などが存在する．Deep-Image-Analogy においては VGG19 の中間特徴量を用いて PatchMatch を行い，画像間の画素ごとの密な対応を取ることができる．この画像間の対応を用いることで，スタイルの転写を行うことができる．Neural Best-Buddies においては Deep-Image-Analogy の手法をベースとしており，PatchMatch を行う際に検索対象領域同士の中間特徴量の平均と分散を近づける Instance Normalization を行うことによって，見た目が大きく異なる物体同士の疎な対応関係をより適切に取ることができる．本研究においては画像中の画像中のすべての画素に対して対応関係を取得する必要があるため，があり，疎な対応を取る Neural Best-Buddies は適切ではない．

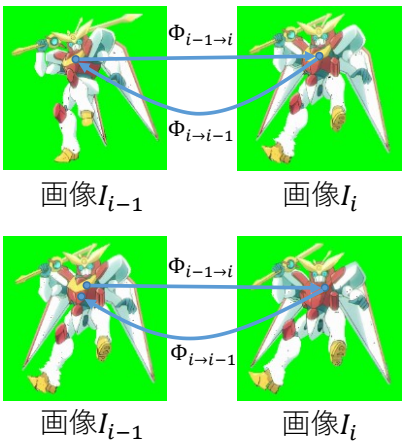

図 1　画素の対応関係のパターン例

3. 提案手法

3.1 対応関係の取り方及びネットワークの選択

本研究では画像間の画素ごとの対応関係を用いることで異常検知を行う．2 つの画像間の対応関係を取る方法として Deep-Image-Analogy [8] を適用する．Deep-Image-Analogy による 2 つの画像 A・B の画素ごとの対応関係を取る．画像 A の座標 $\boldsymbol{a}$ の画素に画像 B の座標 $\boldsymbol{b}$ の画素が対応する場合，画像 B の座標 $\boldsymbol{b}$ の画素の画素値を座標 $\boldsymbol{a}$ の画素にコピーすることで Content が画像 A，Style が画像 B の画像を生成するスタイルの転写を行うことができる．今回はイラストに適用するため，ImageNet で訓練済みの VGG19 の重みではなく，大量のイラスト画像で訓練済みの Illustration2Vec [4] の重みを用いる．Illustration2Vec の重みを利用する際も各中間層の ReLU の出力値を用いて画素ごとの対応関係を取る．

3.2 手法の原理

アニメーションは多数の連続した画像によって構成されており，連続している 2 つの画像は類似している．そのため，連続している 2 つの画像は他の画像を用いるより対応が取りやすいと考えられる．異常検知する画像を画像I_i，画像I_iの 1 フレーム前を画像I_{i-1}とする．また，画像I_iの画素 $\boldsymbol{x}$ に対応する画像I_jの画素を $\Phi_{i\to j}(\boldsymbol{x})$ とする．ここで必ずしも画像I_jの座標 $\boldsymbol{x}$ と $\Phi_{i\to j}(\Phi_{j\to i}(\boldsymbol{x}))$ は一致するとは限らないことに注目する．図 1 に画素の対応関係のパターン例を示す．基本的には図 1 上図ように $\Phi_{i\to i\text{-}1}(\Phi_{i\text{-}1\to i}(\boldsymbol{x}))$は座標 $\boldsymbol{x}$ 付近と

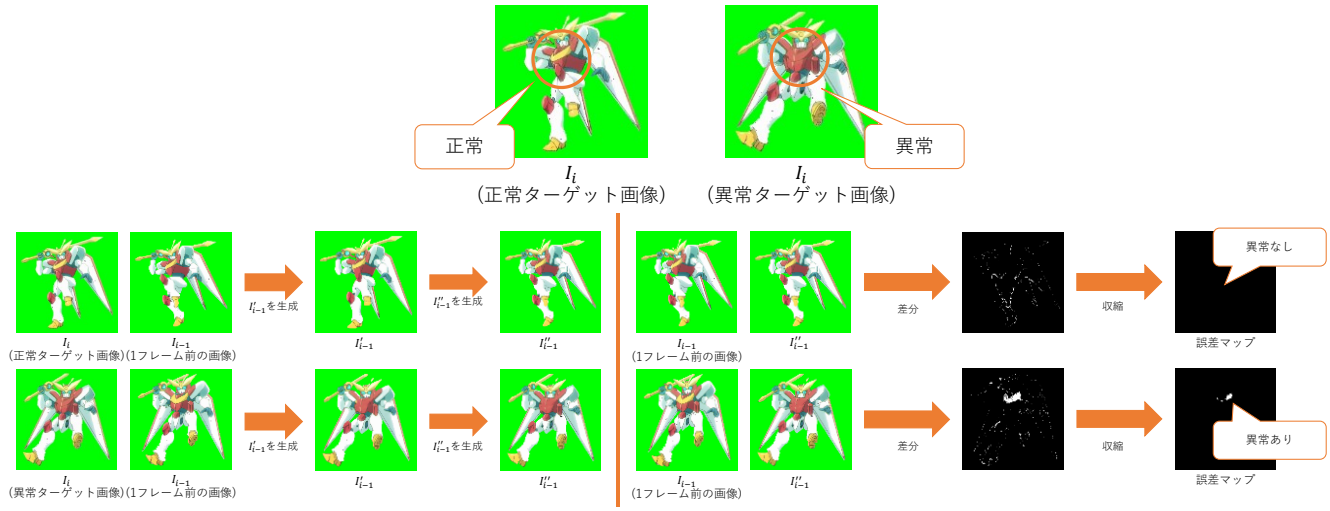

図 2　提案手法の概要

なるが，下図のように $\Phi_{i\to i-1}(\Phi_{i-1\to i}(\boldsymbol{x}))$が座標 $\boldsymbol{x}$ 付近とならないケースも存在した

図 1 の下図のようなケースは画像I_iに色の塗りミス箇所が存在している場合に発生しており，色の塗りミス箇所においてこのような現象が発生している．すなわち，異常画像と正常画像の間では，対応関係の双方向性が失われており，これを利用することによって異常検知を行う．

3.3 異常検知

図 2 に提案手法の概要を示す．画像I_iと画像I_{i-1}から，Content が画像I_i，Style が画像I_{i-1}の画像I'_iを生成する．画像 I の座標 $\boldsymbol{x}$ の画素値を $I(\boldsymbol{x})$ とすると I_{i-1}, I'_iには以下の関係が存在する．

$$I'_i(\boldsymbol{x}) = I_{i-1}\left(\Phi_{i\to i-1}(\boldsymbol{x})\right) \tag{3.1}$$

その後，以下の式(3.2)に基づいて新しい画像I''_{i-1}を生成する．

$$I''_{i-1}(\boldsymbol{x}) = I'_i\left(\Phi_{i-1\to i}(\boldsymbol{x})\right) \tag{3.2}$$

式(3.1)と(3.2)から，I_{i-1}とI''_{i-1}には以下の関係が存在する．

$$I''_{i-1}(\boldsymbol{x}) = I_{i-1}\left(\Phi_{i\to i-1}\left(\Phi_{i-1\to i}\boldsymbol{x}\right)\right) \tag{3.3}$$

先述の通り画像I_iに色の塗りミスが存在しない場合，$\Phi_{i\to i-1}(\Phi_{i-1\to i}(\boldsymbol{x}))$は $\boldsymbol{x}$ 付近となる．よって画像は画像I_{i-1}をほぼ再現すると考えられる．一方，画像I_iに色の塗りミスが存在する場合，画像I''_{i-1}色の塗りミス箇所に対応する場所内の座標 $\boldsymbol{x}$ において，$\Phi_{i\to i-1}(\Phi_{i-1\to i}(\boldsymbol{x}))$は $\boldsymbol{x}$ 付近とならない．そのため，画像I''_{i-1}の中の色の塗りミス箇所に対応する箇所の画素値は，画像I_{i-1}の中の色の塗りミス箇所に対応する箇所の画素値と異なる値となる．よって，色の塗りミス箇所においては画像I_{i-1}と画像I''_{i-1}で色が変化する．以上のことから，画像I_{i-1}と画像I''_{i-1}を比較することによって画像I_iに色の塗りミスが存在するかどうかを判断することができると考えられる．

図 3　作成した作画ミス画像の例

4. 実験

4.1 データセット

実験は 10 のシーンで合計 514 フレームの画像を用いて行った．キャラクターの色の塗りミス箇所を検出するという問題設定から，アニメ画像からキャラクターの部分を切り抜き，背景を緑色にして 256×256 にリサイズするという操作を前処理として行った．実際の制作現場で発生する作画ミス画像はサンプル数が少なく入手することが困難であるため，過去に発生した作画ミスの傾向を基にして疑似的な作画ミス画像を作成した．疑似的な作画ミス画像は，正常な画像の一部のパーツの色を変更することによって作成した．作成した作画ミスの画像の例を図 3 に示す．実際に使用したデータセットは 40 枚の疑似作画ミス画像と 474 枚の正常な画像から構成される．

4.2 実験方法

異常画像の検出は以下の手順通りに画像I_{i-1}と画像I''_{i-1}を比較することによって行った．

1. 以下の式(4.1)を満たす画像 D を生成する．

$$D(\boldsymbol{x}) = \begin{cases} 1 & (|I''_{i-1}(\boldsymbol{x}) - I_{i-1}(\boldsymbol{x})| \geq \tau_1) \\ 0 & (otherwise) \end{cases} \tag{4.1}$$

2. 1.で生成された画像 D を収縮する．
3. ②の操作後に画素値が 1 となっている画素の数が τ_2 以上の場合異常画像とする．

式(4.1)の|·|は $L^*a^*b^*$空間内での絶対値を表す．また，収縮はカーネルサイズ 5×5 で 1 回行われている．

比較を画像I_{i+1}と画像I''_{i+1}にも行い，画像I_{i-1}と画像I''_{i-1}との比較とも合わせて異常判定を行う，どちらかの比較において異常が検出された場合に異常であると判定を行う．

4.3 比較実験

比較実験の対象として AnoGAN [2] を選択する登場する頻度の少ないキャラクターに対して異常検知を行う場合，事前に用意できる画像が少ないという問題設定から，学習に用いる画像をキャラクターごとに 10 枚とした．学習はキャラクターごとに別々に行った．学習の最適化アルゴリズムは Adam，パラメータは l_r=0,0002，β_1=0.5，β_2=0.999，ε=1×10^{-8} と設定し，epoch 数は 500 とした．テスト時に潜在変数を探索する際も Adam (l_r=0,001，β_1=0.5，β_2=0.999，ε=1×10^{-8}) を用い，epoch 数は 500 とした．その後 Anomaly Score を算出し，設定された閾値で異常判定を行った．

4.4 評価

結果から Accuracy，Recall，Precision，F 値を算出する．

Deep-Image-Analogy を用いて意味的な画素対応を取る際に VGG19 ではなく Illustration2Vec の重みを用いる効果も検証するため，VGG19 を使用する場合と Illustration2Vec を使用する場合の ablation study を行う．

品質管理においてミスを漏らしてはいけないという問題設定から Recall が大きいことが望ましいが，同時に手作業でチェックするべき画像の枚数を減らして品質管理を行う人の負担を軽減するという本研究の目的より偽陽性が少ないこと，つまり Precision が大きいことも望ましい．しかし Precision と Recall はトレードオフの関係にある．そのため今回は Precision と Recall の調和平均を取る F 値を最大とするような閾値による実験結果を代表とする．

5. 実験結果

Illustration2Vec を用いた場合の実験結果を表 1 に，VGG19 を用いた場合の実験結果を表 2 に，AnoGAN の実験結果を表 3 に，それぞれの場合の Accuracy，Precision，Recall，F 値を比較したものを表 4 に示す． Illustration2Vec を用いた場合は τ_1=32，τ_2=37 の場合に F 値が最大となり，その値は 0.459 となった．VGG19 を用いた場合には τ_1=1，τ_2=34 の場合に F 値が最大となり，その値は 0.231 となった．AnoGAN の場合は Anomaly Score の閾値を 54600 とした場合に F 値が最大となり，その値は 0.164 となった．Illustration2Vec を用いた場合が最も性能が良く，AnoGAN の結果が最も性能が悪かった．

表 1　Illustration2Vec を用いた場合の結果

τ_1=32　τ_2=37		モデルの予測		合計
		異常	正常	
実際のクラス	異常	17	23	40
	正常	17	457	474
合計		34	480	514

表 2　VGG19 を用いた場合の結果

τ_1=1　τ_2=34		モデルの予測		合計
		異常	正常	
実際のクラス	異常	32	8	40
	正常	205	269	474
合計		237	277	514

表 3　AnoGAN [2] の結果

閾値=54600		モデルの予測		合計
		異常	正常	
実際のクラス	異常	20	20	40
	正常	184	290	474
合計		204	310	514

表 4　Accuracy，Precision，Recall，F 値の比較

	Illustration2Vec を用いた場合	VGG19 を用いた場合	AnoGAN[2]
Accuracy	0,922	0.586	0.603
Precision	0.5	0.135	0.0980
Recall	0.425	0.8	0.5
F 値	0.459	0.231	0.164

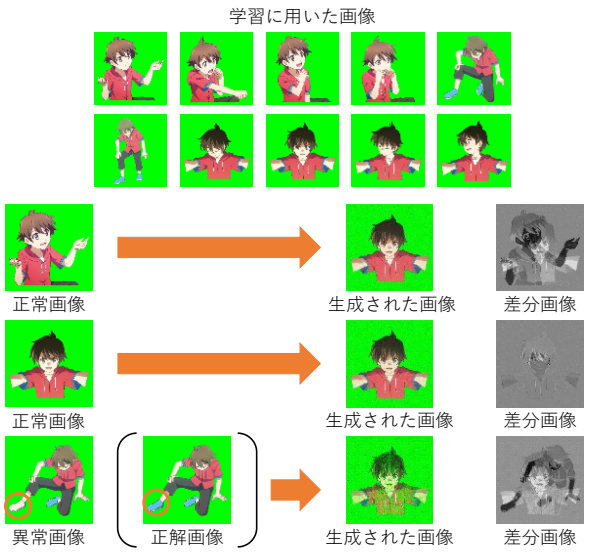

図 4　AnoGAN の結果の概要

図 5　真陽性のパターン

図 6　偽陽性の例

6.　考察

6.1　AnoGAN の結果

AnoGAN の結果の概要を図 4 に示す．図 4 で示されている通り，どのような画像が入力されても同じような画像が出力されている．これは，学習に使用された画像の枚数が少ないためモード崩壊を起こしたためであると考えられる．その結果，入力画像に異常があるかどうかではなく入力画像が「どのような画像が入力されても出力される画像」と近いかによって anomaly score が変化するようになっている．このため，AnoGAN を用いた異常検知は性能が低い結果となっている．

6.2　Illustration2Vec を用いた場合の真陽性の検討

真陽性の画像の中にも 2 つのパターンが存在しており，異常検知の信頼性があるパターンでと信頼性がないパターンに分けられる．図 5 にその 2 つのパターンの具体例を示す, 上の列は異常検知の信頼性があるパターンの例であり，下の列は信頼性がないパターンの例である．

図 5 の上の例は元の画像のネクタイの色を白く塗って疑似的に異常画像とした場合である．生成された画像I''_{i-1}のネクタイの部分も白く塗られており，画像I_{i-1}と画像I''_{i-1}を提案手法で比較すると差分がネクタイの部分で現れる．この場合，ネクタイの部分が異常と判断された上で異常画像と認識されているため信頼性のある真陽性であると言える．このような異常箇所を異常と判断した上で異常画像と判断されている信頼性のある真陽性の画像の枚数は 13 枚であった．

一方図 5 の下の例は元の画像の服の袖の白い部分をピンク色で塗って疑似的に異常画像とした場合である．しかし，生成された画像I''_{i-1}の服の袖の部分は白く塗られていない．また右手や髪飾りの色が変化しており「4.2 実験方法」で提案した方法で比較すると差分がそれらの場所で現れる，

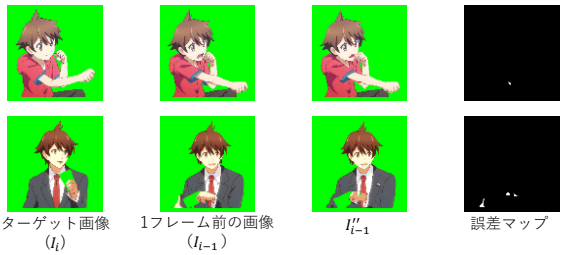

図 7　偽陰性の例 1

図 8　偽陰性の例 2

この場合，服の袖の部分が異常と判断されずに異常画像と認識されているため, 信頼性がない真陽性であると言える．このような異常箇所を異常と判断されずに異常画像と判断されている信頼性のない真陽性の画像の枚数は 4 枚であった．

6.3　Illustration2Vec を用いた場合の偽陽性の検討

偽陽性の例を図 6 に示す，偽陽性は画像I''_{i-1}の生成精度が悪くなるために発生していた．画像I''_{i-1}の生成精度が悪くなるのはアニメーションが激しく変化するフレームであった．

例えば図 6 の上の例においては，画像I_{i-1}ではキャラクターの左腕のひじの部分が右腕で隠されていないが画像I_iでは左腕のひじの部分が右腕で隠されている．そのため，画像I_{i-1}内の左腕のひじの部分から画像I_iに対して適切に画素の意味的な対応関係が取れなくなっており，その結果画像I''_{i-1}では左腕のひじの部分の生成精度が悪くなってしまい，画像I_{i-1}と画像I''_{i-1}を比較した際に異常判定されてしまった．

図 6 の下の例においては，キャラクターの右手が画像I_iにおいてフレームアウトしたり，左手の角度が変わったことにより左手の影の部分が大きく変化したりしているため画像I''_{i-1}で右手や左手の生成精度が悪くなってしまい，異常であると判定されてしまった．

このように画像I_{i-1}内に存在しているものが画像I_i内では遮蔽されてしまう，あるいは存在しない場合に画像I''_{i-1}内においてその部分の生成精度が悪くなってしまい，誤って異常であると判定されてしまう．

6.4　Illustration2Vec を用いた場合の偽陰性の検討

偽陰性には 2 つのパターンが存在した．片方は画像I''_{i-1}において異常箇所に変化が発生しているにも関わらず異常であると判定されなかったパターンであり，もう片方は異常箇所に変化が発生しなかったために異常であると判定されなかったパターンである．異常箇所に変化が発生したパ

図 9　　偽陰性の例 2 における対応関係

ターンを図 7 に，異常箇所に変化が発生しなかったパターンに示す．

異常箇所に変化が発生したパターンにおいては，画像I_{i-1}と画像I''_{i-1}を比較する過程で異常判定が消えていた．図 7 の上図は，キャラクターの左肩の赤いパーツを白く塗った異常画像の異常検知の過程を示したものである．画像I_{i-1}と画像I''_{i-1}の比較の 1.の操作の処理後は異常箇所が残っているが，2.の操作の処理後に異常箇所の面積が異常と判定されないほど小さくなってしまった．つまり，異常箇所の面積が小さいために画像I_{i-1}と画像I''_{i-1}を比較する過程で異常判定となる要素が無くなっていた．図 7 の下図は，キャラクターの左脇のスーツの影を消した異常画像の異常検知の過程を示したものである．この場合スーツの色と影の色が似ているため，画像I_{i-1}と画像I''_{i-1}比較の 1.の操作の処理後に異常箇所が残っていない．つまり，異常箇所の色の変化が小さいために画像I_{i-1}と画像I''_{i-1}を比較する過程で異常判定となる要素が消えていた．

異常箇所に変化が発生したパターンの例における画像I_iと画像I_{i-1}の間の画素の対応関係を図 9 に示す．図 9 に示されている通り，画像I''_{i-1}の中の異常箇所の座標 $\boldsymbol{x}$ の画素の画素値は，画像I_{i-1}の中の座標 $\boldsymbol{x}$ 付近の画素の画素値を参照している．つまり対応関係の双方向性が失われていない．そのため，画像I_{i-1}と画像I''_{i-1}で差分が現れなかった．

7.　　まとめ

本論文では，画像間の意味的な画素の対応を利用することで，1 フレームだけ色の塗りミスが発生している場合の異常検知を行った．Deep-Image-Analogy の手法ベースに Illustration2Vec の特徴量を用いて画素の対応を取ることによって，異常検知の性能が向上した．ただし本手法には以下のような弱点が存在した．

1. アニメーションが激しく変化するようなフレームは，正常であっても異常判定してしまうことがあった．
2. 異常箇所の面積の大きさや色の間違え方によっては，異常画像であっても異常と判定されなかった．

また，図 9 のように，色塗りミスがあるにも関わらず対応関係の双方向性が得られてしまうケースにも適用可能にする必要がある．アニメーションが激しく変化するようなフレームにおける異常検知は，連続するフレーム同士の比較では限界があるため，設定画等の複数枚の正常画像から異常検知をできるようにしたいと考えている．

8.　　謝辞

本研究は，IMAGICA Lab.，JST ACCEL（JPMJAC1602），JSPS 科研費（JP19H01129）の補助を受けています．

参考文献

[1] Dehaene. D, Frigo. O, Combrexelle. S and Eline. P, ”Iterative energy-based projection on a normal data manifold for anomaly localization”, In proc of ICLR 2020.

[2] Schlegl. T, et al., “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery”, In proc of IPMI, 2017.

[3] Krizhevsky. A, Sutskever I, Hinton. G. E, “ImageNet classification with deep convolutional neural networks”, In proc of NIPS, 2012.

[4] Simonyan. K and Zisserman A, “Very deep convolutional networks for large scale image recognition”, In proc of ICLR, 2015.

[5] Saito. M and Matsui. Y, “Illustration2-Vec: a semantic vector representation of Illustrations”, In proc of SIGGRAPH Asia, 2015.

[6] Liu. C, Yuen. J and Torralba. A, “Sift flow: Dense correspondence across scenes and its spplications”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011.

[7] Barnes. C, Shechtman. E, Finkelstein. A and Goldman.D.B, “PatchMatch: A randomized correspondence algorithm for structural image editing”, In proc of SIGGRAPH, 2009.

[8] Lowe. G. E, “Distinctive image features from scale-invariant keypoints”, In proc of IJCV, 2004.

[9] Liao. J, et al., “Visual attribute transfer through deep image analogy”, In proc of SIGGRAPH, 2017.

[10] Aberman. K, et al., “Neural best-buddies: sparse cross-domain correspondence”, In proc of SIGGRAPH, 2018.