# 三次元頂点情報を用いた訓練画像生成による 画像分類精度の向上

十楚 航<sup>1,a)</sup> 山森 一人<sup>2,b)</sup> 相川 勝<sup>3,c)</sup>

概要:

教師あり学習において、異なる視点から見た訓練画像を数多く用意することは非常に煩雑な作業である。 訓練画像の数を増やすために、既存の訓練画像から新たな視点から見た画像を生成する方法が考えられる。 本稿では、三次元モデルを用いて、視点の変化を三次元空間上における三次元モデルの頂点の動きとして 扱い、頂点の動きを訓練画像上に投影することで新たな視点の訓練画像を生成する手法を提案する。提案 手法よって新たに生成した画像を訓練画像に加え、深層学習を通じてその効果を検証する。猫 12 品種を分 類する問題を用いて、提案手法により作成した訓練画像を加えることで、識別精度が向上することを示す。

キーワード:画像分類,画像処理,三次元モデル,分割アフィンワーピング

# Training image synthesis by geometric transformation based on 3D vertex information

WATARU JUSO<sup>1,a)</sup> Kunihito Yamamori<sup>2,b)</sup> Masaru Aikawa<sup>3,c)</sup>

#### Abstract:

It is difficult to prepare the huge number of labeld training images from various viewpoints for image classification. So, it is a nice way to create new training images from original traininig image. In this paper, we propose a method to create new training images from other viewpoint based on 3D object model. Viewpoint shift is expressed by movement of vertexes on 3D object model in three dimentional space, and new training images from other viewpoint are created by projection of the vertex movements into the 2D image. We evaluated efficiency of our method by deep learning.

Keywords: Image classification, Image processing, 3D object model, piecewise affine transformation

# 1. はじめに

近年、機械学習による画像分類の研究が盛んに行われている[1,2]。画像分類は、入力された画像がどのクラス(例えば人、動物、車など)に属しているかを自動で分類する

 宮崎大学 工学研究科 Graduate Engineering, University of Miyazaki, Japan
 宮崎大学 工学教育研究部 Faculty of Engineering, University of Miyazaki, Japan
 宮崎大学 工学部 教育研究支援技術センター Technical Center, Faculty of Engineering, University of Miyazaki, Japan

- a) juso@taurus.cs.miyazaki-u.ac.jp
- b) yamamori@taurus.cs.miyazaki-u.ac.jp
- c) aikawa@taurus.cs.miyazaki-u.ac.jp

問題である。多くの企業が画像分類を様々な問題に応用す ることに取り組んでいる。例えば、近年注目されている分 野の一つに自動運転技術への応用がある。自動運転技術は 前方に映る映像から車線、車、人、標識などを認識し、人 の手を介することなく自動車を運転することを目的として いる。自動車が車線を認識し、その間を自動で走ることで 人がハンドルを握らずとも車線に沿って進むことが可能に なる。また、車や人間などを認識して危険を察知し停車す る技術など、安全面に関しても応用が考えられている[3]。

画像分類を可能にするための機械学習技術の一つとして、 ニューラルネットワーク、中でも畳込みニューラルネット ワーク [4] が非常に注目されている [1,2,5]。畳込みニュー ラルネットワークは、人の視覚システムをヒントにした ニューラルネットワークの一種で、画像認識において非常 に強力な認識器である。ニューラルネットワークなどの機 械学習では、訓練画像から特徴を抽出し、得られた特徴を 学習することによってクラスを識別する。一枚の訓練画像 から取得できる特徴は限られていることから、画像分類で は、多くの、かつ適切な訓練画像を用意することが重要で ある。

しかし、多くの訓練画像を用意することは困難な場合が 多い。例えば、教師あり学習で必要となるラベル付けされ た画像を大量に用意することは容易ではない。なぜなら、 収集した画像が対象とするクラスに本当に属しているかの 判別は、訓練画像を用意する人が行わなければならず、極 めて煩雑な作業となるからである。そのため、用意された 訓練画像から別の訓練画像を生成し、訓練画像自体の数を 増やすことでこの問題を解決するアプローチが提案されて いる[6-9]。

訓練画像生成の従来手法として、アフィン変換やスケー リング等の簡易な幾何変換を用いて、識別対象である画像 内のオブジェクトを変形する手法[6]や、画素値に対して 色調変化を行う手法[7]がある。これら手法は非常に簡易 なため、訓練画像生成に多く用いられる。Paulinら[6]は、 アフィン変換と共にホモグラフィーを用いた画像変形によ る訓練画像生成法を提案している。ホモグラフィーにより 上下左右の4方向に対して幾何変換を行い、視点を擬似的 に変化させた画像を生成する。

アフィン変換やホモグラフィーなどの画像変換手法で は、画像全体に対して一定の処理を行うため、画像上のオ ブジェクトの形状は基本的に変化しないか、変化してもご くわずかである。畳込みニューラルネットワークの畳込み カーネルは画像上の特徴を自動で取捨選択するため、対象 の見え方が異なる、すなわち異なる視点からの画像を多数 準備することが望ましい。オブジェクトの異なる視点から の画像を生成する方法として、対象となるオブジェクトの 三次元モデルを用いて訓練画像を生成する手法が提案され ている。

土屋ら [8] は、人体の三次元モデルを用いて、固定され た監視カメラからの視点に特化した訓練画像を生成する手 法を提案している。人体モデルと仮想カメラを使って、異 なる視点から人体の画像を生成し、背景画像と組み合わせ ることで訓練画像の増加を図る。Javier ら [9] は、あらか じめ仮想空間と三次元モデルを準備し、これらの仮想モデ ルだけを用いて訓練画像を生成する手法を提案した。

三次元モデルを用いて訓練画像を直接生成するこれらの 手法において分類精度を向上させるためには、精巧な三次 元モデルを作成する必要がある。三次元モデルの精度が低 いと、対象とするクラスの特徴が正確に表現できず、訓練 画像の質が低下する。そのため、オブジェクトの輪郭や表 面の質感などの精度を落とさずに画像を生成する方法が必要である。

本稿では、訓練画像と分類対象オブジェクトの三次元モ デルを併用し、視点を変化させた新たな訓練画像を生成す ることで、ニューラルネットワークによる画像分類精度向 上を図る手法を提案する。提案手法では、視点の変化に伴 う三次元モデルの頂点の動きを画像上の物体に投影し、視 点の異なる訓練画像を生成する。

# 三次元モデルを用いた訓練画像生成

三次元モデルは仮想的な三次元空間上において、ポリゴ ンで構成されている。ポリゴンのみを用いた三次元モデル は、オブジェクトの形状しか表すことかできない。そのた め、三次元モデルの表面の質感を与えるためにテクスチャ マッピングが用いられる。テクスチャマッピングによって オブジェクトが持つ色や質感などを表現することができる。

精細な三次元モデルを生成するには、頂点の数を増やし てより多くのポリゴンで構成する、精巧なテクスチャを準 備するなど、モデル生成コストが高くなる。また、動物な どに対しては、テクスチャマッピングでは体毛を精巧に表 現することが難しいため、ファーシェーディングを行う必 要がある。しかし、ファー自身の生成コストや、レンダリ ング時における膨大な計算時間を考えると、訓練画像生成 のために精巧な三次元モデルを作成することは難しい。

この問題を解決するため、三次元モデルから直接訓練画 像を生成する方法ではなく、三次元モデルの形状情報を用 いて訓練画像を生成する方法が考えられる。Yanivら[10] は、三次元モデルを用いて新たな顔訓練画像を生成する手 法を提案している。Yanivらの手法では、学習に使用する 訓練画像が正面顔画像であることを前提に、予め用意した 三次元顔モデルから目や鼻などの特徴となる箇所を選択 し、入力された顔画像を特徴となる点に合致させるように 変形することで新たな正面顔訓練画像を生成する。Yaniv らの手法は、Cootesら[11]の手法に従って構築されてい る。Cootesらの手法は、統計的生成モデルと呼ばれる手法 によって新たなオブジェクトの形状を生成するが、ある程 度決まった形のモデルしか生成できないという問題がある。

### 3. 提案手法

本章では、三次元空間上における三次元モデルの、視 点の変化に伴う頂点の動きを画像上のオブジェクトに投影 して訓練画像を変形する手法について説明する。提案手法 は大きく分けて3つの段階に分かれる。図1に提案手法の 流れを示す。三次元モデルの頂点の動きを画像に投影する ためには、三次元モデルと画像を関連付ける必要がある。 そのため、画像と三次元モデルに対して制御点と呼ぶ点を 設定することで、両者を関連付ける。以降では、画像上の 制御点を2D-制御点、三次元モデル上の制御点を3D-制御 IPSJ SIG Technical Report



図1 提案手法の流れ.



図2 三次元モデルにおける 3D-制御点の設定例.

点と呼ぶ。視点を変化させると、3D-制御点も移動する。こ の三次元空間上での3D-制御点の動きを画像上の2D-制御 点の動きに投影することで、視点が変化した時の画像上の オブジェクトの見かけの変化を表す。2D-制御点の動きに 従って画像を変形することで、視点の変化によるオブジェ クトの見かけの変化を表現した画像を生成する。

#### 3.1 2D/3D-制御点の設定

図2に、三次元モデル上に3D-制御点を設置した例を示 す。3D-制御点は三次元モデルの頂点から構成されている。 3D-制御点には、三次元モデルが持つ特徴的な箇所を選択 する。図2の例では目や鼻、耳など全部で12個の3D-制 御点を設定している。

図3に、画像上に2D-制御点を設置した例を示す。2D-制御点は画像上の1ピクセルからなっている。2D-制御点 は3D-制御点で選択した箇所と同じ箇所とする。これらの 制御点の設定は手動で行っている。

#### 3.2 3D-制御点の見かけの移動量の導出

本節では、視点が変化したときの 3D-制御点の見かけの 変化量を計算する方法を説明する。

視点の変化には、オブジェクトから離脱や接近、並行移 動などが考えられるが、本稿ではオブジェクトを中心とし



図 3 画像上における 2D-制御点の設定例.



図4 視点が a 移動した時の 3D-制御点の見かけの変化の例.

た視点の円周移動を対象とする。オブジェクトの重心を原 点として、3D-制御点が円周上に移動することで視点の変 化を表現する。例として、図4に三次元空間におけるy軸 周りの3D-制御点の円周移動を示す。

移動前の *i* 番目の 3D-制御点 *P<sub>i</sub><sup>3D</sup>* の座標を式 (1) で表す。 制御点の y 軸周りの回転を考える場合、 y 座標の値は変わ らないため定数とする。

$$\begin{aligned} \boldsymbol{P}_{i}^{3D} &= (x_{i}^{3D}, y_{i}^{3D}, z_{i}^{3D}, 1)^{T}, \\ &= (r_{xz} sin\theta, y_{i}^{3D}, r_{xz} cos\theta, 1)^{T}. \end{aligned} \tag{1}$$

同様に x 軸周りの移動前の 3D-制御点の座標を式 (2) で 表す。制御点の x 軸周りの回転を考える場合、x 座標の値 は定数とする。

$$P_{i}^{3D} = (x_{i}^{3D}, y_{i}^{3D}, z_{i}^{3D}, 1)^{T},$$
  
=  $(x_{i}^{3D}, r_{yz} cos\theta, r_{yz} sin\theta, 1)^{T}.$  (2)

3D-制御点  $P_i^{3D}$  を、現在の角度 $\theta$ から角度 $\alpha$ だけ回転させる。このとき、新しい3D-制御点の位置を $P_i^{3D'}$ とする。 3D-制御点の移動は軸を中心とした円運動であるため、新しい3D-制御点の位置は式(3)で表すことができる。

$$\boldsymbol{P}_{i}^{3D'} = \boldsymbol{R} \boldsymbol{P}_{i}^{3D}. \tag{3}$$

このとき、R はある軸に従う回転行列である。y 軸周りの

3D-制御点の円運動の場合、回転行列 **R**<sub>y</sub> は式 (4) で表される。

$$\boldsymbol{R}_{y} = \begin{pmatrix} \cos \alpha & 0 & -\sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$
 (4)

同様に、*x* 軸周りの 3D-制御点の回転行列 *R<sub>x</sub>* は式 (5) で 表すことができる。

$$\boldsymbol{R}_{x} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$
 (5)

式 (3) により 3D-制御点  $P_i^{3D}$  は新しい位置  $P_i^{3D'}$  に移動 する。このとき、3D-制御点の移動量  $D_i^{3D}$  は式 (6) で表す ことができる。

$$D_i^{3D} = P_i^{3D'} - P_i^{3D}.$$
 (6)

この 3D-制御点の移動量を三次元空間上での見かけの変化 量として扱う。

#### 3.3 見かけの移動量の画像への投影

3.2 節で三次元空間における 3D-制御点の見かけの変化 量を求めた。しかし、この移動量は三次元空間上の移動量 であるため画像へ直接反映できない。本節では、三次元空 間上で求めた制御点の移動量を二次元空間上の移動量に変 換する手法について説明する。また、変換した移動量に従 い 2D-制御点を移動させ、新たな訓練画像を生成する方法 についても説明する。

3.3.1 スケーリング

三次元空間上のオブジェクトと画像上のオブジェクトの 大きさは異なるため、3D-制御点の移動を 2D-制御点の移 動へと直接投影することはできない。そのため、三次元空 間上の 3D-制御点の移動量を画像上の 2D-制御点の移動量 へとスケーリングする。

オブジェクトのサイズを合わせるために、画像上のオブ ジェクト、三次元モデルの両方において、制御点で囲まれ たオブジェクトの幅と高さを使用する。図5は画像上に設 置した2D-制御点の中からx座標、y座標それぞれの最大、 最小の座標値を持つ2D-制御点を選出した例である。図中 では2D-制御点を赤い点で表現し、大きな赤い点を2D-制 御点中の最大、最小の座標値を持つ2D-制御点として表し ている。

2D-制御点で囲まれたオブジェクトの  $x 座標の幅は x_{max}^{2D} - x_{min}^{2D}$ 、 $y 座標の幅は y_{max}^{2D} - y_{min}^{2D}$ で求められる。同様に、三次元モデル上の 3D-制御点に対しても、オブジェクトの幅と高さを求める。



図 5 2D-制御点で囲まれたオブジェクトの x 座標の幅、y 座標の幅 の例.

次に、画像、三次元モデルのオブジェクトが持つそれぞ れの幅からスケール因子を求める。式(7)、式(8)により、 *x*軸方向、*y*軸方向それぞれのスケール因子*S<sub>x</sub>、S<sub>y</sub>*を求 める。

$$S_x = \frac{x_{max}^{2D} - x_{min}^{2D}}{x_{max}^{3D} - x_{min}^{3D}},$$
(7)

$$S_{y} = \frac{y_{max}^{2D} - y_{min}^{2D}}{y_{max}^{3D} - y_{min}^{3D}}.$$
(8)

#### 3.3.2 三次元上での現在の角度の導出

式 (3) を用いて、 y 軸を中心とした円周上を視点が移動し た時の 3D-制御点の新たな位置を求めるためには、3D-制 御点と z 軸との現在の角度 θ と視点の移動角度 α が必要で ある。そのため、現在の 3D-制御点の角度 θ を重心からの 位置によって決定する。3D-制御点の現在の位置  $P_i^{3D}$  は、 画像上の対応する 2D-制御点の重心からの位置を三次元に 投影することで求める。

図6は、ある3D-制御点 $P_i^{3D}$ の、y軸からみた現在の位置を示している。図6中の3D-制御点 $P_i^{3D}$ の現在の角度 $\theta$ は、 $x_i^{3D} \geq z_i^{3D}$ の値が分かれば求めることができる。しかしながら、画像上の2D-制御点の位置から求めることのできる座標値は x 座標のみで、z 座標の値を求めることはできない。これは、画像自体に奥行き(z 軸の情報)がないためである。3D-制御点の現在の角度を求めるための z 座標以外の要素を考えると、回転軸と3D-制御点を結ぶ半径 r を求めることができれば、現在の角度 $\theta$ を求めることができることが分かる。なぜなら、三次元空間上の特定の軸に対して 3D-制御点の回転を考えたとき、回転軸から 3D-制御点からなる半径 r と、画像上の 2D-制御点から得られた $x_i^{2D}$ 、 $y_i^{2D}$ の値を用いて 3D-制御点の現 在の角度 $\theta$ を求める手順を説明する。

2D-制御点の重心は式 (9) で求めることができる。ここで *N* は制御点の総数を表す。



図 6 3D-制御点 P<sub>i</sub><sup>3D</sup> と重心との位置関係.

$$C = (x_c^{2D}, y_c^{2D}),$$
  
=  $\frac{1}{N} \left( \sum_i x_i^{2D}, \sum_i y_i^{2D} \right).$  (9)

y 軸周りの 3D-制御点の回転を考える場合、3D-制御点は xz 平面上を移動する。そのため、xz 平面上において y 軸と 3D-制御点からなる現在の角度  $\theta_i^{xz}$  は式 (10) で求めること ができる。

$$\theta_i^{xz} = \sin^{-1} \left( \frac{x_i^{2D} - x_c^{2D}}{r_{xz} S_x} \right).$$
(10)

同様に、*x* 軸周りの 3D-制御点の回転を考える場合、3D-制御点は *yz* 平面上を移動する。そのため、*yz* 平面上にお いて *x* 軸と 3D-制御点からなる現在の角度  $\theta_i^{yz}$  は式 (11) で 表すことができる。

$$\theta_i^{yz} = \cos^{-1} \left( \frac{y_i^{2D} - y_c^{2D}}{r_{yz} S_y} \right).$$
(11)

3.3.3 2D-制御点の移動量の導出

3.2 節で述べたように、ある 3D-制御点が角度  $\alpha$  だけ回転したときの移動量  $D_i^{3D}$  は式 (6) で求めることができる。本節では、式 (6) により求まる三次元上の移動量  $D_i^{3D}$  を、画像上の移動量  $D_i^{2D}$  に変換する手順を説明する。

はじめに、式(6)で求めた3D-制御点の移動量を、画像上の2D-制御点の移動量に変換する。変換は式(7)、式(8)で 求めたスケール因子を用いて以下の式で表すことができる。

$$\boldsymbol{D}_i^{\boldsymbol{x}2\boldsymbol{D}} = \boldsymbol{S}_{\boldsymbol{x}} \boldsymbol{D}_i^{\boldsymbol{x}3\boldsymbol{D}},\tag{12}$$

$$\boldsymbol{D}_{i}^{y2D} = \boldsymbol{S}_{y} \boldsymbol{D}_{i}^{y3D}.$$
 (13)

2D-制御点の最終的な位置は、y軸を軸とした回転移動の 場合は式(14)、x軸を軸とした回転移動の場合は式(15)で 表すことができる。



図7 2D-制御点の移動.

$$P_i^{2D'} = P_i^{2D} + D_i^{x2D}, \tag{14}$$

$$P_i^{2D'} = P_i^{2D} + D_i^{y2D}.$$
 (15)

これらの工程をすべての 2D-制御点に対して行い、2D-制御点の新たな位置を決定する。図 7 では元の 2D-制御点 を赤色で、新しい位置の 2D-制御点を青色で示している。 3.3.4 変換行列の決定

前節までに、すべての 2D-制御点毎に、視点の移動に伴 う新しい位置を求めた。本節では、移動前の 2D-制御点と 移動後の 2D-制御点を用いて、2D-制御点に囲まれた領域 毎に異なる変換行列を求める手順について説明する。オブ ジェクトの領域毎に適切な変形を行うことで、画像の自然 な変形を目指す。

はじめに、画像上のオブジェクトの 2D-制御点に囲まれ た領域毎の変形を可能にするために、 $P_i^{2D}$ を組み合わせて 三角形のパッチを生成する。パッチの作成にはドロネー三 角分割法 [12] を用いる。視点移動前の 2D-制御点の組み合 わせによる *k* 番目のパッチを式 (16) で表す。

$$\boldsymbol{T}_{k} = \left(\boldsymbol{P}_{l}^{2D}, \boldsymbol{P}_{m}^{2D}, \boldsymbol{P}_{n}^{2D}\right).$$
(16)

また、新しい位置に移動した同じ制御点の組み合わせに よるパッチを式 (17) で表す。

$$T_{k}' = \left(P_{l}^{2D'}, P_{m}^{2D'}, P_{n}^{2D'}\right).$$
(17)

つぎに、分割されたパッチ毎にアフィン変換を行う。オ ブジェクトをパッチに分割し画像を変形する手法を、分割 アフィンワーピング [13] という。図8 にパッチを配置した 結果を示す。赤色の三角形パッチは元の 2D-制御点からな る *T*<sup>k</sup> である。

 $T_k$ から $T'_k$ への変形は式 (18)で表すことができる。

$$\boldsymbol{T}_{\boldsymbol{k}}^{'} = \boldsymbol{H}_{\boldsymbol{k}}\boldsymbol{T}_{\boldsymbol{k}} \tag{18}$$

各パッチ毎に変換行列  $H_k$ を求めることで、パッチごとに異なる変形を行うことができる。



図8 画像を三角形分割した例.



図9 提案手法による訓練画像変換の例.

## 3.3.5 変換画像の作成

図9に、提案手法によって画像を変形した例を示す。この例では猫を対象とし、頭部を提案手法により変形している。図9は、原画像を角度0度とし、左から順に-20、-10、0、10、20度に回転した画像である。

# 4. 評価実験

本章では、生成した画像を訓練画像に加えることでニュー ラルネットワークの分類精度が向上するかどうかを検証 する。実験に使用する訓練画像はオックスフォード大学の Visual Geometry Group が公開している The Oxford-IIIT Pet Dataset [14] を使用する。このデータセットの中の猫 12 ク ラス、画像 2,400 枚を対象に学習を行う。学習に使用する訓 練画像は各クラス 50 枚、テスト画像は訓練画像を含まない、 各クラス 50 枚とする。テスト画像は訓練画像を含まない、 名クラス 50 枚とする。テスト画像に関しては、オリジナル の画像のみを使用する。学習プラットフォームは NVIDIA 社が提供する NVIDIA DIGITS を用いる。NVIDIA DIGITS では機械学習ライブラリ caffe を使用している。また、学 習に使用するニューラルネットワークは Google が提供し ている GoogLeNet を使用する。

#### 4.1 実験条件と評価方法

4 つの訓練画像セットを各々ニューラルネットワークに 学習させ、学習済みのニューラルネットワークを用いてテ スト画像の正答率を評価する。ニューラルネットワークに 学習させる訓練画像セットは以下の4つを準備する。 Set 1. オリジナルの訓練画像のみ

Set 2. オリジナルの訓練画像と、提案手法により y 軸に対

衣 1 子宮ハフメータ				
	Set 1	Set 2	Set 3	Set 4
訓練画像	600	1,800	1,800	3,000
の枚数				
テスト画	600			
像の枚数				
学習方法	Stochastic gradient descent (SGD)			
学習係数	0.01			
SGD の学	200			
習回数				

して-10,10 度の視点移動を表現した訓練画像

- Set 3. オリジナルの訓練画像と、提案手法により y 軸に対 して-20,20 度の視点移動を表現した訓練画像
- Set 4. オリジナルの訓練画像と、提案手法により y 軸に対して-10,-20,10,20 度の視点移動を表現した訓練画像 正答率は式 (19) で定める。

正答率(%)=

<u>正しいクラスに分類されたテスト画像の数</u>×100(%) すべてのテスト画像の数 (19)

評価では、学習終了後のニューラルネットワークに対し てテスト画像を入力したときの正答率の 10 回の平均値を 用いる。

表1に本実験で使用する訓練画像の枚数、ニューラル ネットワークのパラメータを記す。各訓練画像セットの画 像はGooLeNetの入力層の次元に従い、256×256に正規化 する。NVIDIA DIGITS の GoogLeNet の画像変換パラメー タ flip と crop はオフとする。同時に、中間画像の使用フラ グもオフとする。これらの設定により、学習は SGD と訓 練画像のみによって行われるため、訓練画像の追加が正答 率に与える影響を調べることができる。

#### 4.2 評価と考察

実験により、提案手法によって生成した訓練画像を学習 セットに加え、訓練画像を増加させることによるニューラ ルネットワークの正答率について調査する。

図10に、学習回数に対する各訓練画像セットでの正答率 を示す。図10から、元の訓練画像セットに提案手法によ り生成した訓練画像を加えることで正答率が向上している ことが確認できる。図10より、Set1とSet2、Set3、Set 4の正答率を比較すると、2%から4%の向上が得られてい る。このことから、擬似的な視点変換による訓練画像生成 により識別精度を向上させることができることが分かる。 また、学習が進むにつれて正答率は向上し、約100epochで ほぼ変化しなくなる。このとき、訓練画像に対する正答率 はほぼ100%であり、学習回数は十分であると言える。

図 11 は、擬似的な視点変換によって作成した訓練画像を



図10 各訓練画像セットにおける平均正答率の遷移.



図11 新たに識別できたテスト画像の枚数.

加えることで、新たに識別できるようになったテスト画像 の枚数を示している。比較対象とした訓練画像セットは、 最も正答率の低い Set 1 と最も高い Set 3 とした。カウント するテスト画像は、10 個の学習済みニューラルネットワー クにテスト画像を入力した際、一度でも識別できた画像の 枚数と定義している。図 11 から、アビシニアン種では 10 枚、エジプシャンマウ種の識別などでは 5 枚の正答画像の 増加をみた。また、ベンガル種については正答した画像枚 数について変化はない。ボンベイ種、シャム種については 正答画像の減少をみた。

アビシニアン種とエジプシャンマウ種で識別画像が増え た理由は、使用している GoogLeNet の畳込みカーネルが 5×5や7×7などの小さなフィルターによって画像を認識 しているため、訓練画像の少しの変化でも異なる特徴を捉 えることができたためだと考えられる。図12に、新たに 識別できたアビシニアン種のテスト画像を、図13に、新 たに識別できたベンガル種のテスト画像を示す。

提案手法により生成した訓練画像を追加することで誤認 識が増加した品種について、ボンベイ種はメインクーン種 に、シャム種はラグドール種とメインクーン種に分類され る事例が見られた。ボンベイ種は体毛が漆黒であることが 特徴的であるが、照明の具合により頭部の模様が視認で きる画像があり、これが誤認識に繋がった可能性がある。 シャム種は顔部分の黒色が特徴であるが、ラグドール種も よく似た体色であり、これが誤認識に繋がった可能性が考



図 12 新たに正しく識別できたアビシニアン種のテスト画像.



図 13 新たに正しく識別できたベンガル種のテスト画像.

えられる。シャム種とメインクーン種の体色は大きく異 なっており、体色以外の特徴が誤認識に繋がった可能性が 高いと考えられるが、畳込みニューラルネットワークでは 画像上のどの部分が識別の鍵として使われているか分から ないため、今後さらに検討する必要がある。

図 14 は、GoogLeNet の画像変換パラメータ flip と crop フラグをオンにして、各訓練画像セットを学習させたと きの、3回の学習の平均正答率の推移を示す。GoogLeNet の flip は元画像を y 軸中心に反転し、crop は訓練画像から 224×224 の画像を切り抜き、新たな訓練画像として使用す る。flip と crop は、各学習段階の訓練画像入力時にランダ ムに行われるため、提案手法と組み合わせることで、より 多くの訓練画像が生成されることになる。

提案手法により作成した訓練画像と、flip および crop を 加えた訓練画像セットの正答率を図 14 に示す。図 14 か ら、元の訓練画像セットよりも高い正答率を示しているこ とが分かる。このことから、提案手法は他の訓練画像生成 法とも併用できることが分かる。すなわち、従来手法の訓 練画像生成法と提案手法を組み合わせることで、より高い 正答率が得られることが分かった。

以上の実験から、提案手法によって擬似的な視点変換を



図14 従来手法と組み合わせた平均正答率の推移

表現した訓練画像を訓練画像セットに加えることで、猫12 クラス識別の正答率の向上を図ることができた。畳込み ニューラルネットワークは与えられた訓練画像から特徴を 自動的に抽出しているため、画像上のどの部分を選択して 学習しているかわからない。しかし、実験の結果から正答 率の向上が確認できることから、元の訓練画像セットと生 成した訓練画像セットとの微小な差を学習している可能性 は指摘できる。

#### 5. おわりに

本稿では、オブジェクトが持つテクスチャの情報を保持 しつつ見かけの変化を表現するために、三次元モデルの頂 点情報を用いて三次元的な見かけの変化に基づき画像を変 形する手法を提案し、画像分類精度の向上を図った。

提案手法により新たに作成した訓練画像を用いて、ニュー ラルネットワークの画像分類精度の向上を検証した。実験 の結果、提案手法により生成した訓練画像を加えることで 猫 12 クラスの分類精度の向上を図ることができた。提案 手法によって訓練画像を増やしたことにより、畳込みカー ネルが異なる特徴を取得できたためと考えられる。一方、 生成した訓練画像の追加により、正答率が減少する場合が あることが分かった。訓練画像の正答率は 100%となって いたため、訓練画像に最適化された過学習によって正答率 の減少が引き起こされた可能性が考えられる。

今後の課題は、より汎用的な画像変形方法と、既存手法 と組み合わせた、更なる認識率向上のための訓練画像生成 法の考案が挙げられる。

#### 参考文献

- Krizhevsky, A., Sutskever, I. and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems* 25 (Pereira, F., Burges, C., Bottou, L. and Weinberger, K., eds.), Curran Associates, Inc., pp. 1097–1105 (2012).
- [2] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A.: Going Deeper with Convolutions, *CoRR*, Vol. abs/1409.4842 (2014).

- [3] Google: Google Self-Driving Car Project, Google (online), available from (https://www.google.com/selfdrivingcar/) (accessed 2016-1-6).
- [4] Cun, Y. L. L., Bottou, L., Bengio, Y. and P.Haffner: Gradient-Based Learning applied to Document Recognition, *Proceedings of IEEE*, Vol. 86, No. 11, pp. 2278–2324 (1998).
- [5] Lin, M., Chen, Q. and Yan, S.: Network In Network, *CoRR*, Vol. abs/1312.4400 (online), available from (http://arxiv.org/abs/1312.4400) (2013).
- [6] Paulin, M., Revaud, J., R.Harchaoui, Perronnin, F. and Schmid, C.: Transformation Pursuit for Image Classication, *CVPR 2014* (2014).
- [7] Wu, R., Yan, S., Shan, Y., Dang, Q. and Sun, G.: Deep Image: Scaling up Image Recognition, *CoRR*, Vol. abs/1501.02876 (2015).
- [8] 土屋成光,山内悠嗣,藤吉弘亘:人検出のための生成型
   学習と Negative-Bag MILBoost による学習の効率化,画
   像の認識・理解シンポジウム (MIRU2012) (2012).
- [9] Marín, J., Vázquez, D., Gerónimo, D. and López, A. M.: Learning appearance in virtual scenarios for pedestrian detection, *CVPR*, IEEE Computer Society, pp. 137–144 (2010).
- [10] Taigman, Y., Yang, M., Ranzato, M. and Wolf, L.: DeepFace: Closing the Gap to Human-Level Performance in Face Verification, *CVPR*, IEEE, pp. 1701–1708 (2014).
- [11] Butakoff, C. and Frangi, A. F.: Multi-view face segmentation using fusion of statistical shape and appearance models, *Computer Vision and Image Understanding*, Vol. 114, No. 3, pp. 311–321 (2010).
- [12] Bradski, G. and A., K.: 詳解 OpenCV.
- [13] Solem, J.: Programming Computer Vision with Python: Tools and Algorithms for Analyzing Images, Oreilly and Associate Series, O'Reilly Media, Incorporated (2012).
- [14] Parkhi, O. M., Vedaldi, A., Zisserman, A. and Jawahar, C. V.: Cats and Dogs, *IEEE Conference on Computer Vision and Pattern Recognition* (2012).