Convolutional-Recurrent Neural Network による自己運動識別

神谷龍司† 川口俊樹† 福井宏† 石井育規‡ 小塚和紀‡ 羽川令子‡
築澤宗太郎‡ 山下隆義† 山内悠嗣† 藤吉弘亘†
†中部大学 ‡パナソニック株式会社
E-mail: {shinryu@vision.cs, yuu@isc, yamashita@cs, hf@cs}.chubu.ac.jp

Abstract

Convolutional-Recurrent Neural Network (C-RNN) は，Deep Convolutional Neural Network (DCNN) による特徴抽出と Recurrent Neural Network (RNN) による時系列対応により，動画像などの時系列を学習することができる．C-RNN は，畳み込み層と LSTM 層を End-to-End で学習するため，DCNN と RNN のパラメータを同時に最適化する必要があり，パラメータがうまく学習されない場合がある．本研究では，1 人称視点映像からの自己運動識別を対象とし，C-RNN のより良いパラメータを学習する方法及び入力データの形式について検討する．学習は，DCNN と RNN を End-to-End で学習する方法と，DCNN と RNN を別々に学習する 2 段階学習を導入する．入力には，カラー画像，オプティカルフローおよびオプティカルフローの方向ベクトルの可視化画像，カラー画像と方向ベクトルの可視化画像の 3 種類の入力データを用いる．方向ベクトルの可視化画像を用いることで，各画素の移動方向や強度を 1 枚の画像で表現できる．評価実験の結果，方向ベクトルの可視化画像を 2 段階で学習した場合に最も良い自己運動識別精度を得ることができた．

1 はじめに

Deep Convolutional Neural Network(DCNN) [2] は，一般物体認識の性能を大きく上回ったことで注目されており，シーンセグメンテーションや歩行者検出等の様々な物体認識で高い性能を達成している．しかし，DCNN は 1 枚のカラー画像またはグレースケール画像を入力とした場合に高い性能を発揮できるが，自己運動識別のような，時系列情報を必要とする認識問題に対しては十分な性能を発揮できない．ニューラルネットワークにおいて，時系列情報を入力して認識する RNN [3] があり，動画像認識や自然言語処理，音声認識など時系列情報を扱うような問題設定で用いられている．RNN は，時系列に従ってサンプルを入力する構造となっており，時刻 t の中間層には時刻 t のサンプルと時刻 $t-1$ の中間層の応答値が入力される．学習時には，Back Propagation Through Time(BPTT) や Long Short Term Memory(LSTM) [11] を使用し，時系列を遡りながらパラメータを更新する．RNN は，BPTT や LSTM を用いることで時系列対応が可能となり，高精度な時系列パターンの学習が可能となる．

DCNN と RNN をベースとした C-RNN は，DCNN による特徴抽出と RNN による時系列対応により高精度な時系列パターンの学習が可能である [1]．しかし，C-RNN は End-to-End で学習するため，DCNN と RNN のパラメータを同時に最適化する必要があり，問題設定によって性能が発揮されない場合がある．また，入力サンプルに動画のフレームを用いた場合，1 枚のフレームから物体の動きを推定するのは困難である．

本研究では，C-RNN による自己運動識別を対象として，学習における最良なパラメータ更新方法，及び入力データの形式について検討する．学習では，DCNN と RNN を End-to-End で学習する方法と，別々に学習する 2 段階学習を比較する．入力データの形式には，カラー画像，オプティカルフロー及びオプティカルフロー方向ベクトルの可視化画像，カラー画像と方向ベクトルの可視化画像で検証する．オプティカルフローの可視化画像は，1 枚のカラー画像で数フレームの物体の移動量を表現できる．そのため，DCNN のように 1 枚の画像から特徴量を抽出して認識する方法でも，自己運動識別に有効な特徴量を獲得できる．本稿では，自己運動識別に対して学習法や入力データの形式を検討することにより，C-RNN の推定性能を向上させる．

2 DCNN と RNN

本章では，C-RNN のベースとなる CNN と RNN の構造及び関連研究について述べる．

2.1 Deep Convolutional Neural Network

DCNN は，一般物体認識において高い性能を発揮しており，ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 で高い認識性能を実現して以降，注目されている [7]．ILSVRC2012 以降，一般物体認識をはじめ家の番号認識 [4]，シーン認識 [5]，物体検出 [6]

など様々な認識問題で用いられ，高い認識性能を達成している．

DCNN では，入力データにカラー画像やグレースケール画像が与えられる．そして，入力データを畳み込み層に入力して，$k_x \times k_y$ の重みフィルタを畳み込む．入力データに対して重みフィルタを畳み込んだ際に得られる応答値は，活性化関数を通して次層の特徴マップに格納する．活性化関数には，シグモイド関数や Rectified Linear Unit(ReLU)[7]，Maxout[8] などが一般的に用いられる．その後，プーリングにより特徴マップをダウンサンプリングする．プーリングには，Max Pooling が一般的に用いられており，2×2 程度の局所領域から最大値を出力する．DCNN では，畳み込み層とプーリング層が階層的に配置されており，繰り返し積み重ねることで深いネットワークを構築している．畳み込み層とプーリング層を積み重ねた後に，全結合層が配置される．全結合層は，特徴マップを 1 次元の特徴ベクトルに変換して，全結合層へ入力される．入力された特徴ベクトルは，結合重みとの内積をとり，活性化関数を通して次層へ入力される．全結合層を繰り返し配置した後に，出力層が配置される．出力層では，問題設定に合わせて出力層を構成することで，様々な識別問題に対応することができる．自己運動識別により前進，後退，右回転，左回転を識別する場合，出力層のユニット数は 4 つとなる．そして，活性化関数には softmax 関数を用いる．softmax 関数は，各クラスに対するスコアを出力し，出力されたスコアが最大となるクラスを識別結果として出力する．

2.2 Recurrent Neural Network

RNN は，動画像や音声などの時系列データを扱うことができるニューラルネットワークである．Sutskever らは，RNN を機械翻訳に適応し，英語の文章をフランス語へ翻訳する手法を提案している [9]．また，RNN は画像認識においても応用されており，ネットワークに入力された画像が，どのような画像であるかを文章で説明する手法も提案されている．

RNN の入力層では，動画像や音声などの時系列データが，時刻 $t = 1$ から時系列に従ってサンプルが入力される．時刻 t の中間層には，時刻 t の入力層の応答値と時刻 $t-1$ の中間層の応答値が入力される．このように，連鎖的にサンプルを入力し，時刻 $t-1$ の中間層の応答値を時刻 t の中間層に入力することで，時系列に対応した認識が可能となる．RNN の学習では，LSTM [11] が一般的に用いられる．RNN で用いられる BPTT は，時系列を遡るようにパラメータを修正するが，取り扱う時系列が長くなるに連れて誤差の勾配が爆発的に高くなったり，消失する問題が発生する．LSTM は，図 1 のように中間層のユニットをメモリユニットに置き換え，RNN のパラメータを更新する．ここで，メモリユニットには入力ゲート，忘却ゲート，出力ゲートがある．入力ゲートは，ユニット U2, U3, U4, U6 と結合されており，これらのユニットは時刻 t の中間層に入力される応答値となる．入力ゲートでは，時刻 t の中間層に入力する応答値を制御している．忘却ゲートは，U1 ユニットに対して過去の記憶をどの程度忘却させるかを調節している．忘却ゲートが 0 に近い値である場合は過去の記憶を忘却し，1 に近い場合は過去の記憶を残して次のメモリユニットへ伝播される．出力ゲートでは，最終的なユニットの出力を外部に伝達するかしないかを制御している．これにより，学習誤差を算出する際に，不必要な誤差を算出しないようにしている．LSTM は，3 つのゲートを用いて効率的に誤差を伝播することができるため，BPTT と比べて長期の時系列データを扱うことができる．そのため，遠い過去の記憶を RNN の出力に反映することが可能となるため，高精度な認識が可能となる．

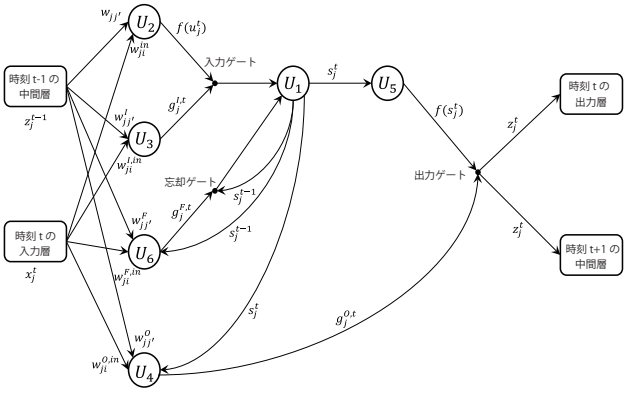


図 1　LSTM の構造

3 C-RNN による自己運動識別

C-RNN[1] は，DCNN と RNN を組み合わせたニューラルネットワークである．DCNN による動画像を対象とした特徴抽出と，RNN による時系列情報を用いた学習ができる．C-RNN の学習方法は 2 つある．1 つ目は，DCNN と RNN を End-to-End に学習する方法，2 つ目は DCNN と RNN を 2 段階に学習する方法である，本章では，C-RNN の 2 つの学習方法について述べる．

3.1 End-to-End の学習

C-RNN の構造は，下位層に DCNN の畳み込み層とプーリング層，上位層に RNN の中間層と出力層を配置した構造となっている．ここで，RNN の中間層に対して LSTM が導入されている．C-RNN に学習データを入力する場合，時刻 $t = 1$ から時系列に従って学習データを入力する．時刻 t における LSTM が導入されている層には，時刻 $t-1$ における C-RNN の中間層の応答値と時刻 t の畳み込みとプーリングの応答値が入力さ

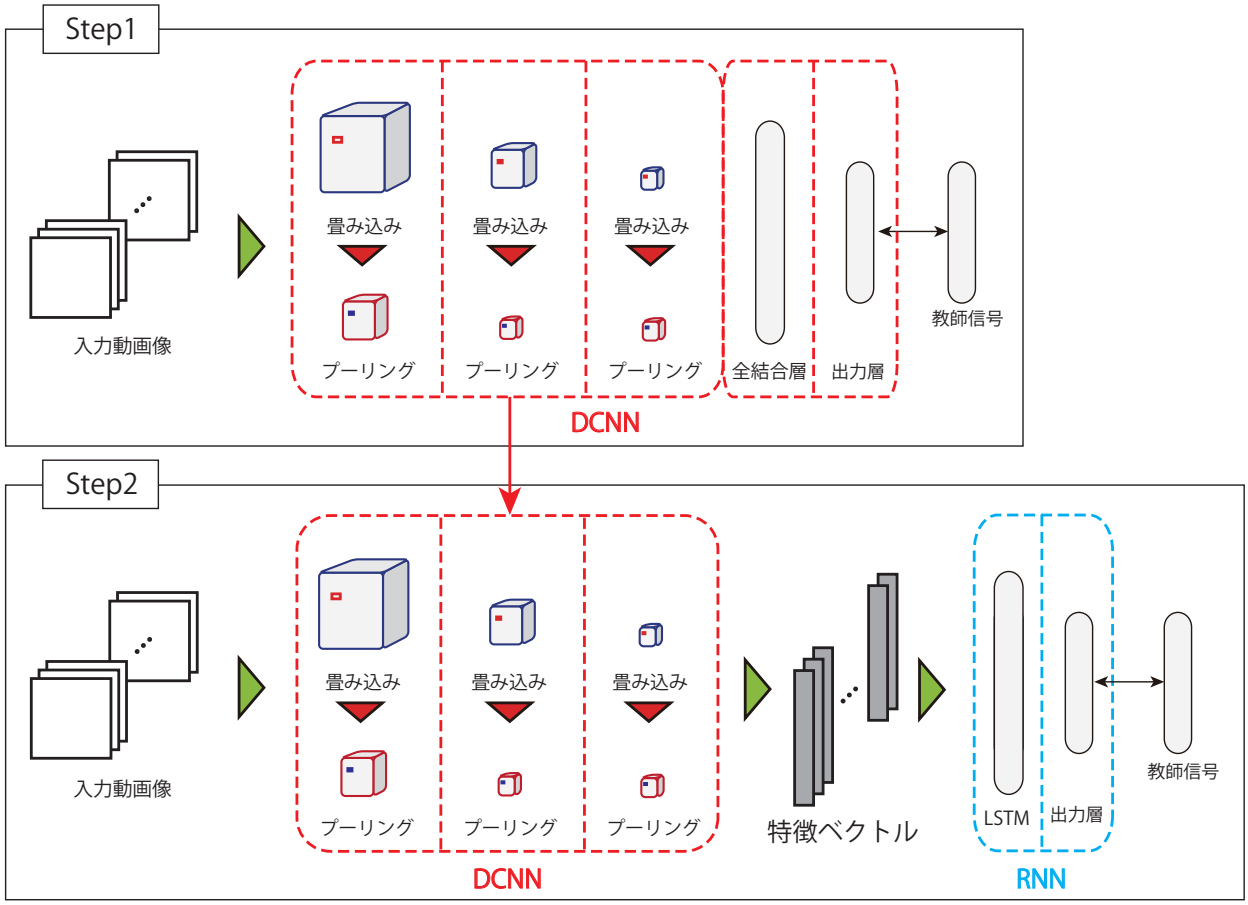


図 2　2 段階学習の流れ

れる．そして，各時刻の C-RNN の出力を得たとき，各時刻における C-RNN の学習誤差を算出する．算出した各時刻の学習誤差から，BPTT により C-RNN のパラメータを更新する．

End-to-End で C-RNN を学習させた場合，BPTT によりネットワーク全体が時系列を捉えれる特徴量を獲得できるように学習されるため，高精度な識別が可能となる．

3.2　2 段階学習

C-RNN の End-to-End による学習は，動画像を順伝播させて誤差を算出し，LSTM により DCNN と RNN のパラメータを更新する．しかし，DCNN と RNN のパラメータを同時に更新するため，問題設定によってパラメータがうまく学習されない場合がある．C-RNN の学習方法において，DCNN と RNN を別々に学習する 2 段階学習がある．パラメータの更新を別々に行うことで，1 フレームから自己運動識別に有効な特徴量を DCNN により抽出し，RNN により時系列に対応した学習と識別が可能となる．

C-RNN の 2 段階学習の流れを図 2 に示す．Step1 では，通常の DCNN に 1 フレーム毎に入力し，誤差逆伝播法により学習する．そして，Step1 で学習した DCNN と RNN を用いて C-RNN を構築する．Step1 で学習した DCNN は，畳み込み層及びプーリング層を，RNN の下位層へ結合させる．C-RNN を構築した後に，RNN と同様に時系列に従いながら入力データを C-RNN に入力する．ここで，畳み込み層及びプーリング層により得られる特徴マップは，1 次元の特徴ベクトルに変換され，RNN へ入力する．学習によるパラメータの更新では，LSTM により RNN のパラメータのみ更新される．

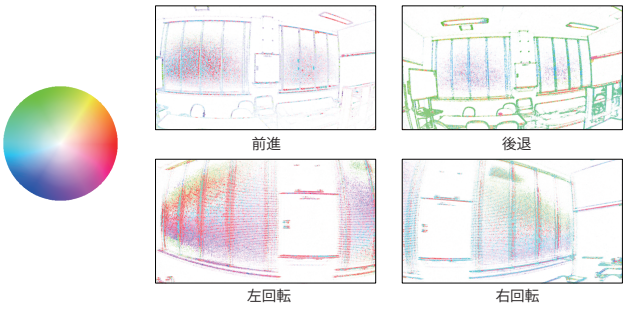


図 3　方向ベクトルの可視化例

3.3　入力データの方式

入力データには，カラー画像を入力する方法とオプティカルフローの方向ベクトルの可視化画像を入力する方法を考える．カラー画像の場合，動画像中から 1 フレームを取り出して，ネットワークに入力する．しか

し，1 枚のカラー画像からシーン中の物体の動きを捉えるのは，非常に困難である．そのため，シーン中の物体の動きを捉えるためにオプティカルフローから得られる方向ベクトルを用いる．オプティカルフローの方向ベクトルを用いることで，フレーム間の物体の動きを 1 枚の画像で表現できる．そのため，学習により特徴量を獲得する際に，物体の動きのみに着目しながら学習できる．また，DCNN のように 1 枚の画像から自己運動識別する方法に対しても有効である．

オプティカルフローの方向ベクトルを用いる場合，はじめに前後 2 フレームからオプティカルフローにより方向ベクトルを算出する．そして，算出した方向ベクトルから，方向ベクトルの向きを色，強度を明るさで変換した可視化画像を生成する．ここで，図 3 に方向ベクトルの可視化画像の例を示す．図 3 の生成した方向ベクトルの可視化画像は，DCNN や RNN，C-RNN のネットワークに入力される．学習時は，連続する 2 フレームから方向ベクトルの可視化画像を 1 人称視点の映像から生成して学習データセットを構築し，ネットワークを学習する．識別時は，連続する 2 フレームから方向ベクトルの可視化画像を生成してネットワークに入力し，自己運動識別する．

4 評価実験

C-RNN の学習方法及び入力データの方式の有効性を調査するために評価実験を行う．比較手法として，C-RNN(2 段階学習)，C-RNN(End-to-End)，DCNN，RNN の 4 つの手法を用いる．

本実験では，一人称視点で撮影された動画像から，カメラ装着者の運動を識別し，精度を評価する．推定する運動は前進，後退，右回転，左回転の 4 パターンとする．自己運動は，屋内で撮影し，動画像の各フレームには，前進，後退，右回転，左回転の 4 方向の教師データが付与されている．また，各手法では Data Augmentation は行わずに学習する．学習及び評価に使用するフレーム数を表 1 に示す．ここで，フレームのサイズは 90×160 のカラー画像を使用する．

表 1　使用するカメラ映像のフレーム枚数

サンプル	前進	後退	右回転	左回転	合計
学習	5,200	5,000	5,300	4,900	20,400
評価	500	500	600	500	2,100

また本実験では，カラー画像，方向ベクトルの可視化画像，カラー画像と方向ベクトルの可視化画像の 3 種類の入力データを用いる．可視化画像を作成する際に使用する方向ベクトルの算出手法には，Gunnar Farneback 法 [12] を用いる．オプティカルフローは，前後の 2 フレームから抽出する．本実験に使用する DCNN は，表 2 のように畳み込み層が 3 層，全結合層が 2 層の構造である．RNN は，表 3 のように 3 層の RNN を用いる．C-RNN は，表 2 の畳み込み層と表 3 の中間層及び出力層が，結合した構造となる．学習パラメータは，学習率は 0.01，ミニバッチのサイズは 10，更新回数は 10 万回である．

表 2　DCNN のネットワーク構造

入力層		$3 \times 90 \times 160$
1 層目	重みフィルタ	$32 \times 3 \times 3$
	ReLU	-
	MaxPooling	2×2
2 層目	重みフィルタ	$64 \times 3 \times 2$
	ReLU	-
	MaxPooling	2×2
3 層目	重みフィルタ	$64 \times 2 \times 2$
	ReLU	-
	MaxPooling	2×2
4 層目	ユニット数	64
	Dropout	50%
5 層目	ユニット数	128
	Dropout	50%
6 層目	ユニット数	4
	Softmax	-

表 3　RNN のネットワーク構造

入力層		$3 \times 90 \times 160$
1 層目	ユニット数	64
	Dropout	50%
	LSTM	-
2 層目	ユニット数	128
	Dropout	50%
	LSTM	-
3 層目	ユニット数	4
	Softmax	-

4.1 自己運動識別精度の比較

各手法による識別精度を表 4 に示す．表 4 より，C-RNN と DCNN 及び RNN を比較すると，入力データに方向ベクトルの可視化画像を用いたほうが精度が高いことがわかる．方向ベクトルの可視化画像は 1 枚のフレームに映っている各物体の動きを捉えることができるため，自己運動識別の精度が向上したと考えられる．

DCNN と RNN の識別精度を比較すると，RGB 画像を用いた場合は RNN のほうが精度が良く，可視化画像を用いた場合においては DCNN のほうが精度が良い．これを踏まえて C-RNN を End-to-End で学習した場合と 2 段階で学習した場合を比較すると，RGB 画像を

表 4　各手法による推定精度

手法	RGB 画像	可視化画像	RGB 画像 + 可視化画像
C-RNN(2 段階学習)	30.71	**84.52**	61.00
C-RNN(End-to-End)	**49.66**	82.66	**73.57**
DCNN	28.47	**81.52**	48.90
RNN	**35.71**	71.71	**67.57**

用いた場合は End-to-End の学習のほうが精度が良く，可視化画像を用いた場合は 2 段階学習が良い．これは，RNN が適している問題は End-to-End の学習を用いるほうが良く，DCNN が適した問題であれば 2 段階学習を用いるほうが良い傾向があるといえる．RGB 画像+可視化画像においても同様であり，RNN の識別精度が DCNN よりも高いため End-to-End の学習方法を用いるほうが 2 段階学習より識別精度が高い．また，入力データにおける精度を比較すると，RGB 画像よりも可視化画像を用いた場合のほうが圧倒的に精度が良いことがわかる．これは，DCNN により特徴ベクトルを抽出する際に，RGB 画像を用いた場合は画像中から移動方向や強度といった情報は得られないが，可視化画像を用いることで DCNN の畳み込みに適した入力となったため，識別精度が向上したと考えられる．

4.2　自己運動識別の結果

図 4(a), (b), (c) に，方向ベクトルの可視化画像を入力とし，2 段階学習した C-RNN の自己運動識別結果を示す．図 4(a) は，室内で撮影した動画像に対して自己運動識別した結果である．図 4(a) より，方向ベクトルの可視化画像は物体の動きを 1 枚のフレームで捉えることができるため，高精度に自己運動識別できることが確認できる．

次に，自動車走行環境において自己運動識別を行った際の推定結果を示す．ここで，自己運動識別に使用する C-RNN は，図 4(a) で推定に使用したネットワークを用いる．実験では，昼の時間帯に撮影した走行シーンと夜間に撮影した走行シーンを用いて，自己運動識別を行う．図 4(b) に，昼の時間帯に撮影したシーンで自己運動識別を行った結果を示す．図 4(b) より，学習で使用したデータと異なった場合においても，自己運動識別が可能であることが確認できる．方向ベクトルの可視化画像を用いることで，シーン中の物体の動きに着目して学習により特徴量を獲得できる．そのため，学習サンプルと異なるシーンを評価した場合でも，自己運動識別が可能となる．図 4(c) に，夜間に撮影したシーンで自己運動識別を行った結果を示す．カラー画像では，夜間のシーンにおいて物体が映らなくなるため，自己運動識別が非常に困難となる．方向ベクトルの可視化画像は，夜間でも図 4(c) のように方向ベクトルを抽出できる．そのため，図 4(c) のような夜間のシーンにおいても物体の動きを捉えることができ，自己運動識別が可能である．

5　おわりに

本稿では，自己運動識別を対象として，C-RNN における最良な学習方法と，入力データの形式について検討を行った．1 人称視点の動画像を用いた自己運動識別では，C-RNN を用いることで DCNN と RNN を単体で用いるよりも高精度な識別が可能となった．また，C-RNN の学習において，RNN による時系列対応が効果的であれば End-to-End の学習が適しており，DCNN による特徴量設計のほうが効果的であれば 2 段階学習が適していることが分かった．入力データの形式は，RGB 画像よりもオプティカルフローにより算出した方向ベクトルの可視化画像を用いることで推定精度が向上した．方向ベクトルの可視化画像を用いて学習した C-RNN を用いることで，学習サンプルと大きく異なるシーンにおいても自己運動識別が可能であり，夜間のシーンにおいても自己運動識別が可能になる．

参考文献

[1] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen, ”Convolutional Recurrent Neural Networks:Learning Spatial Dependencies forImage Representation”, Computer Vision and Pattern Recognition, 2015.

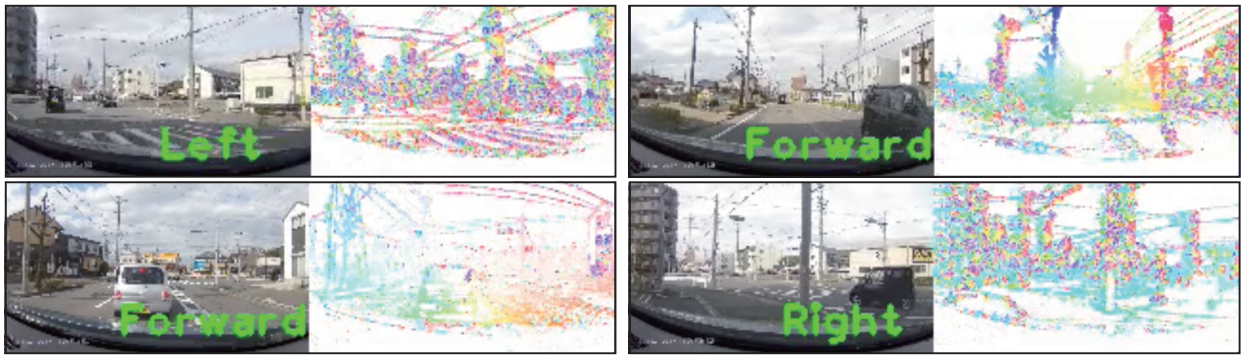
[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ”Gradient-Based Learning Applied to Document Recognition”, Proceedings of the IEEE, pp. 2278-2324, 1998.

[3] J.Elman, ”Finding Structure in Time”, Cognitive Science, Vol. 2, No.14, pp.179-211, 1990.

[4] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnold, and V. Shet, ”Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks” ,CoRR, Vol. abs/1312.6082, 2013.

[5] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, ”Learning Hierarchical Features for Scene

(a) 室内の認識結果



(b) 自動車走行環境の認識結果 (昼)

RGB 画像　　可視化画像　　RGB 画像　　可視化画像

(c) 自動車走行環境の認識結果 (夜)

図 4　自己運動識別結果の例

Labeling", IEEE Transacations on Pattern Analysis and Machine Intelligence, 2012.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation" , Computer Vision and Pattern Recognition, 2014.

[7] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks", Neural Information Processing Systems, 2012.

[8] I. Goodfellow, D. W. Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks", arXiv preprint arXiv:1302.4389, 2013.

[9] l. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks", Neural Information Processing Systems, 2014.

[10] P.Werbos, "Generalization of backpropagation with application to a recurrent gasmarket model", Neural Networks 1 (4): 339-356, 1988.

[11] S.Hochreiter, "LONG SHORT-TERM MEMORY", Neural Computation 9(8) 1735-1780, 1997.

[12] G.Farneback, "Two-Frame Motion Estimation Based on Polynomial Expansion", IN SCIA2003, pp.363-370, 2003.