Deep Convolutional Neural Network による手形状領域の抽出

山下 隆義† 綿末 太郎‡ 山内 悠嗣† 藤吉 弘亘†
†中部大学 ‡とめ研究所
E-mail: yamashita@cs.chubu.ac.jp

Abstract

ディープラーニングは音声認識や物体認識などの様々な分野で高い汎化性が注目されている．本稿ではディープラーニングの１つである Deep Convolutional Neural Networks を手形状の領域抽出に利用する．手形状は同一クラス内のバリエーションが大きい．そのため，学習には大量の学習サンプルが必要となるが，人手で収集することは大変困難である．そこで，限られた学習サンプルから学習サンプルを生成する方法を利用する．また，提案手法では maxout などのディープラーニングの過学習を防ぎ，汎化性を向上させる手法を用いる．これにより，提案手法は学習サンプルの生成を行わない場合と比べて，領域抽出精度を向上させることができた．

1 はじめに

ディープラーニングは音声認識や物体認識を含む多くのコンペティションに用いられている．特に，Deep Convolutional Neural Networks (ConvNets) は手書き文字認識 [4]，番地認識 [12]，標識認識 [11]，物体認識 [13] などの多くのベンチマークでトップレベルの性能を達成している．Krizhevsky らの ImageNet2012 コンペティションでの勝利や，Zeiler らの ConvNets を認識問題だけでなく位置検出タスクへの応用と ImageNet2013 での勝利を通じて，ConvNets を含むディープラーニングはより一層，一般的な機会学習手法となっている．

ConvNets の利点は各タスクに適した特徴を入力画像の画素をそのまま利用して簡単に抽出できる点である．これにより人手による特徴設計が不必要となり，特徴設計の負担を軽減することができる．また，ネットワーク全体でより複雑な特徴を抽出することが可能となる．

一方，ConvNets は過学習に陥りやすいことや学習時間がかかることなどの欠点がある．多くの研究者がこれらの ConvNets の欠点を解決するための手法を提案しているが，多くの学習サンプルが必要となる問題が残っている．上述のベンチマークでは大量の学習サンプルが用意されているが，ConvNets のポテンシャルを引き出すために十分なサンプル数とは言えない．対象とする物体のクラス内バリエーションが多い場合は特に大量の学習サンプルが不可欠となる．この問題は物体認識だけでなく，物体の領域抽出のタスクにおいても同様である．そこで，本稿では領域抽出において，限られた学習サンプルから大量のサンプルを自動生成させて抽出精度を向上させる方法を提案する．領域抽出の対象として，クラス内のバリエーションが多い手の形状とする．

2 関連研究

1990 年代前半から多くの研究者が ConvNets を用いた手法を提案している [3][4]．近年では，ConvNets は手書き文字認識 [4] をはじめ，テキスト検出 [8] や画像認識 [11][13][16]，人体検出 [22][18][19] など幅広く利用されている．ConvNets は 1000 クラスの物体認識タスクで従来手法よりも圧倒的な性能を達成してブレークスルーを起こして以降，Sermanet らが ConvNet の多大な可能性を人体検出で示している．この ConvNets は階層的な構造とともに，１層目の出力を直接３層目に利用するスキップ構造を併用している．また Sparse coding による教師なしの事前学習も利用している．この手法では，グローバルな形状情報と局所的な情報の両方を特徴としている．一方，ConvNets による領域抽出は Jain らにより位置検出に用いられている [7]．この手法では，領域矩形の代わりに物体の輪郭を利用して物体の位置を学習している．

多くの研究者が過学習を防ぐ方法や学習プロセスの問題を解決する方法を提案している．特に，テストサンプルに対する汎化性を向上させる方法として，max pooling や average pooling などの pooling[9][10] や maxout[15] がある．pooling は同じ特徴マップ内の近傍領域における最大値を出力として選択する方法であり，maxout は異なる特徴マップ間で同じ位置にある値のうち最大値を出力として選択する方法である．Pooling と maxout の両方とも位置の不変性があることが知られている．また，局所解に陥ることを防ぐための事前学習方法として Auto encoder がある．Auto encoder は ConvNets を構成する各層のパラメータを教師なし

学習により事前に初期値を決める方法であり，全層のパラメータを同時に決めるのではなく，１層ごとに階層的に決めていく方法である．また，全結合層のパラメータを学習する際に，一定の結合のパラメータを０とする方法であり，過学習を防ぐ方法として知られている．

一方，ConvNets の構成による過学習を防ぐ方法以外に，サンプル生成により学習サンプルを十分に用意する方法がある．最も簡単なサンプル生成は位置ずれさせたサンプルを用意する方法や画像反転である [13][14]．また，スケールや回転を適用する方法もある [21]. Simard らはさらに高度な方法として Elastic distortion と呼ばれる変形方法を利用して文字認識での効果を示している [5]．我々は Elastic distortion の効果を領域抽出のタスクでも同様にあることを本稿にて示す.

また，大量の学習サンプルを利用して学習する場合，学習時間が大幅にかかる．そのため，GPU による並列処理がよく使われている．Krizhevsky らは GPU のバス幅を考慮した ConvNets の構成を提案し，ConvNets による性能向上とともに高速化を実現している．近年は，Theano[20] などにより，ConvNets の並列学習環境が整っており，容易に学習環境を構築することができる．本稿では Theano を利用して ConvNets を構成させる．

3 Convolutional Neural Network

ConvNets は Hubel ら [1] による視覚野の局所受容野にもとづく畳み込み層とサンプリング層，すべてのユニットを結合する全結合層および最終的な認識結果を出力する認識層から構成されている．また，入力層として画像の画素値やエッジまたは正規化した値を入力する層もある．畳み込み層は $Kx \times Ky$ サイズの M 個のカーネルをもち，入力に対してカーネルの畳み込みを行い特徴マップを出力する．特徴マップに対して，サンプリング層ではサブサンプリングを行う．Scherer ら [9] は，サブサンプリング手法の１つである max pooling が学習の収束が早いことや汎化性が高いことを示している．また，Boureau ら [10] は pooling によるサブサンプリングの有効性を理論的に解析している．Max pooling は同じ特徴マップ内の 2×2 などの一定領域における最大値を出力する．Deep learning では，畳み込み層とサンプリング層を繰り返し複数層重ねる構成となっている．全結合層では，これらの構成から得られた値を重み付き和として全結合して各ユニットの値を求める．そして認識層では全ユニットの値を特徴ベクトルとし，Soft max により各クラスの確率を求める．ConvNets の各パラメータはランダムに決めた初期値から誤差逆伝播法による教師あり学習で更新していき，最適なパラメータを得る [2] [6].

3.1 ConvNets の学習

ConvNets の学習には誤差逆伝播法が用いられる．誤差逆伝播法は式 (1) および式 (2) に示すように，勾配降下法により誤差 E が最小となるような結合重みを推定する．

$$E = \frac{1}{2}\sum_{p=1}^{P} E_p \tag{1}$$

$$w_{ji}^{(l)} \leftarrow w_{ji}^{(l)} + \Delta w_{ji}^{(l)} = w_{ji}^{(l)} - \lambda \frac{\partial E_p}{\partial w_{ji}^{(l)}} \tag{2}$$

ここで，$\{p|1, ..., P\}$ は学習サンプル，$\boldsymbol{o_p}$ は学習サンプル p に対する出力，$\boldsymbol{t_p}$ は学習サンプル p の教師ラベルである．λ は学習率，$w_{ji}^{(l)}$ は l 番目の層のユニット i から次層のユニット j に対する結合重みである．各学習サンプルの誤差 E_p は出力とラベルとの差を累積である．$\Delta w_{ji}^{(l)}$ は式 (3) のように表すことができる．

$$\Delta w_{ji}^{(l)} = -\lambda \delta_k^{(l)} y_j^{(l-1)} \tag{3}$$

$$\delta_k^{(l)} = e_k \phi(V_k^{(l)}) \tag{4}$$

$$V_k^{(l)} = \sum_j w_{kj}^{(l)} * y_j^{(l-1)} \tag{5}$$

$y_j^{(l-1)}$ は，$(l-1)$ 番目の層のユニット j の出力，e_k はユニット k の誤差，$V_k^{(l-1)}$ は $(l-1)$ 番目の層の全ユニットから l 番目の層のユニット k への結合値の累積である．局所的な勾配降下は式 (4) から得ることができる．活性化関数 ϕ は様々なバリエーションがあり，シグモイド関数や ReLu[13] などがある．ネットワーク全体の結合重みはあらかじめ決めた回数または収束条件を満たすまで繰り返し更新される．

誤差逆伝播法による誤差 E の求め方は，*full-batch*, *online* and *Mini-batch* の複数の方法がある．*Full-batch* はすべての学習サンプルを同時に与えて，結合重みを更新する．この方法の場合，繰り返し回数は少ないが勾配が大きくなり収束しにくい．一方，*Online* は学習サンプルを１つづつ与えて更新するため，勾配は小さくなるが繰り返し回数が膨大となる．*Mini-batch* はこれらの中間的な方法で学習サンプルを小さなサブセットに分けて更新する．*Mini-batch* は大量の学習サンプルを利用する場合でも効率的に重みを更新することができ，よく用いられている．

4 提案手法

4.1 学習サンプルの生成

実環境でのアプリケーションに ConvNets を利用する場合，単純な背景下での限られたバリエーションの学習

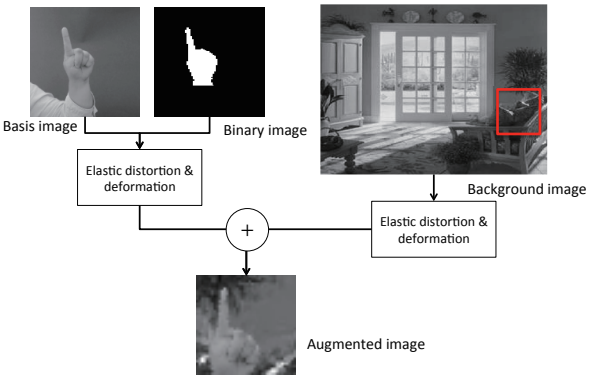

図 1　学習サンプルの生成方法.

表 1　Deformation range of each factor.

deformation factor	range
translation	± 3
scaling	± 5%
rotation	± 5
brightness	± 10%

サンプルだけでなく，様々な背景下におけるサンプルが必要となる．我々はそれらのサンプルを用意する方法として，図 1 に示すような data augmentation と背景合成によるデータ生成方法を利用する．最初に，グレースケールの学習サンプルとともにその画像のバイナリのラベルデータを用意する．そして，Elastic distortion により学習サンプルとラベルデータを変形させる．Elastic distortion では，各画素の位置を新たなターゲットの位置 x^* に配置する．新たな位置 x^* は次式のようにランダムな変位とスケーリング係数 α から求める．

$$\boldsymbol{x}^* = \boldsymbol{x} + \alpha\Delta\boldsymbol{x}. \tag{6}$$

$\Delta\boldsymbol{x}$ は-1 から 1 の間のランダムな実数である．Simard らの手法のように，$\Delta\boldsymbol{x}$ は標準偏差 σ の正規分布を畳み込んで得られた平滑化変位である [5]．位置 $\boldsymbol{x}^*$ の値はバイリニア補間を用いて周辺画素の値から求められる．

学習サンプルとバイナリラベルデータに Elastic distortion を適用後，背景画像と合成を行う．その際，表 1 に示すような大きさ，並進，回転の変形を加える．背景領域はオリジナルの背景画像から位置や大きさをランダム決めて切り取る．

4.2　ネットワーク構成

提案手法のネットワーク構成を図 2 に示す．ネットワークは入力層，畳み込み層，サブサンプリング層，全結合層，バイナリ層の 5 つの種類から構成されている．Max pooling は学習の収束を早くすることができ，また汎化性を向上させることができる．提案手法では，max pooling に次いで maxout を利用する．Max pooling は，同じ特徴マップ内の近傍から最大値を選択する一方，maxout は異なる特徴マップ間における同じ位置の最大値を選択する．Max pooling は特徴マップのサイズが半分になるが，maxout は特徴マップ数が半分になる．全結合層は前層の出力を特徴ベクトルとして受け取り，dropout を利用して学習する．バイナリ層は全結合層が出力する特徴ベクトルを受け取り，バイナリ画像を出力する．バイナリ層の各ユニットは画素に相当しており，各画素が 1 となる確率を出力する．

4.3　入力層

ネットワークは過学習を防ぐために大量の学習サンプルを利用して学習する．しかしながら，大量の学習サンプルを収集することは困難である．4.1 節で述べたように Elastic distortion を利用することで限られた学習サンプル数から大量のデータを生成することが可能である．入力層には生成したグレースケールの学習サンプルを与える．

4.4　畳み込み層

我々は畳み込み層の活性化関数として，maxout を用いる [15]．従来のシグモイド関数のような活性化関数や ReLU[13] と比べて高い表現力がある．従来の活性化関数は式 (7) に示すような h_i in $\boldsymbol{h} = (h_1, ...h_i, i \in I)$ として定義される．

$$h_i = \sigma(x^T W_i + b_i). \tag{7}$$

$\sigma(\cdot)$ はシグモイド関数であり，x は入力ベクトル，W_i は結合重み，b_i はバイアス項である．Maxout は式 (8) に示すように，複数の特徴マップから最大値を選択する．

$$h_i = \max_{j\in[1,k]} z_{ij}, \tag{8}$$

$$z_{ij} = x^T W_{ij} + b_{ij} \tag{9}$$

4.5　全結合層

全結合層のユニットは，前層のすべてのユニットと重み付き結合で結ばれている．全結合層の学習には dropout を利用する．Dropout は過学習を防ぎ，汎化性を向上させる効果的な方法である [14]．Dropout はランダムに選んだ結合重みを 0 とし，残りの結合重みのみ更新する方法である．一般的には一回の更新で全結合重みのうち 50%を 0 にする．

4.6　バイナリ層

領域抽出のタスクのためにバイナリ層を利用する．バイナリ層は全結合層に基づいており，全結合層と同様，前層のすべてのユニットと結合する．バイナリ層の出力ユニット数は入力画像のサイズと同じである．各出力ユニットは抽出したい領域らしさを確率として出力する．

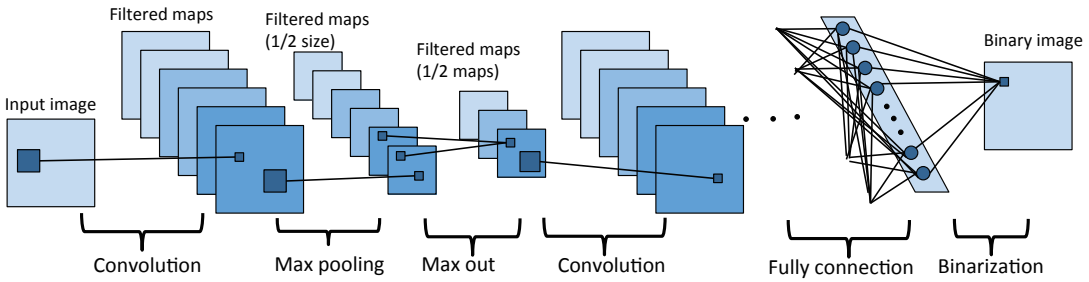

図 2　ネットワークの構成.

表 2　ネットワークの構成

layer	type	size,# of kernels
input	grayscale	40×40
1st	convolution	5×5, 32
2nd	max pooling	2×2
3rd	maxout	4
4th	convolution	5×5, 32
5th	max pooling	2×2
6th	maxout	4
7th	fully connected	200
output	L2 norm	1600

4.7　ネットワークの学習

畳み込むカーネルの要素，結合重みおよびバイアスのすべてのパラメータを $\boldsymbol{W}$ とし，それらはランダムな値を初期値とする．パラメータは誤差逆伝播法により逐次更新されていき，$t+1$ 回目に更新するパラメータ $\boldsymbol{W_{t+1}}$ は式 (10) のように更新される．

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \epsilon_{t+1}\frac{\partial E}{\partial \boldsymbol{W}} \tag{10}$$

認識問題の場合，Softmax が一般的に用いられるが，我々は式 (11) のような L2 ノルムを利用する．

$$E = ||\boldsymbol{h} - \boldsymbol{y}||^2 \tag{11}$$

学習率は式 (12) に示すように，繰り返し回数に応じて小さくしていく．

$$\epsilon_{t+1} = \frac{\epsilon_0 \tau}{\max(t, \tau)} \tag{12}$$

ϵ_0 は初期の学習率，τ は学習率を減少させていく回数である．

5　実験

領域抽出タスクにおける ConvNet および学習サンプルの自動生成の効果を示すために，同一クラス内のバリエーションが多い手形状を対象とした領域抽出精度の比較実験を行う．評価データセット例を図 3 に示す．

図 3　評価データセット

表 3　領域抽出の性能

手法	precision	recall	F 値
提案手法	0.9033	0.9234	0.9133
画像生成なし	0.8958	0.9008	0.8983

手形状として 6 種類の形状を含んでおり，各形状には個人差や傾きなどの変形もある．また，背景は実環境であり，照明変化も生じている．学習サンプルは 1600 枚のグレースケール画像およびバイナリラベルデータをベースとし，提案手法は Elastic distiortion により 200 万枚の画像を生成する．画像生成を行わない手法では，背景画像との合成のみ行い，同様に 200 万枚を学習に利用する．mini batch による学習の更新回数は 20 万回とする．

5.1　実験結果

表 3 に画像生成の有無による precisin-recall 率を示す．これより，画像生成を行った提案手法の方が画像生成なしと比べて性能が良いことがわかる．すなわち，学習サンプルの枚数が同一の場合でもバリエーションの豊富さが重要となっている．また，バイナリ層の出力を可視化した例を図 4 に示す．入力がグレースケールが像にも関わらず提案手法は複雑な背景下でも手領域を抽出できていることがわかる．さらに，提案手法は背景の複雑さだけでなく照明変化にも頑健となっている．

6　考察

6.1　更新回数による性能比較

学習時の誤差の推移を図 5，および提案手法の更新回数による性能の変化を表 4 に示す．図 5 より 4000 回更

図 4　複雑背景下からの手領域の抽出結果

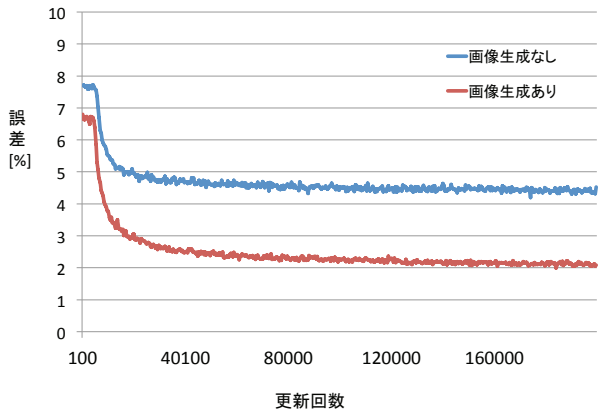

図 5　提案手法の学習時における誤差の推移

新する間に誤差は大きく減少しており，それ以降徐々に収束していることがわかる．また，画像生成ありの方が誤差が大きく減少している．表 4 からも precision および recall が 5 万回までに大きく向上していることがわかる．5 万回以降の更新回数ではわずかであるが，F 値についても改善されている．これらより，画像生成することで学習サンプルのバリエーションが多くなっているため，多様な変化に頑健になっていると言える．

6.2　カーネルサイズによる性能比較

次に，カーネルサイズによる性能比較を行う．性能の比較結果を図 5 に示す．これより 5×5 の場合が最も F 値が大きくなっていることがわかる．しかしながら，他のカーネルサイズと比較して差はそれほど大きくない．これより，カーネルサイズが性能に及ぼす影響は小さいと言える．

7　Conclusion

本稿では，ConvNets をベースとした領域抽出手法を提案した．提案手法は全結合層をベースとしたバイナリ層を利用している．また，入力画像はグレースケールにも関わらず，複雑な背景下でも手領域を抽出することができている．今後は他の形状への応用および，他手法との比較を行う．

表 4　提案手法における更新回数による性能比較

更新回数	precision	recall	F 値
0	0.2302	1.0000	0.3742
50000	0.8930	0.9135	0.9032
100000	0.8968	0.9199	0.9082
150000	0.9054	0.9186	0.9120
200000	0.9033	0.9234	0.9133

表 5　提案手法におけるカーネルサイズによる性能比較

更新回数	precision	recall	F 値
3x3	0.9097	0.9133	0.9115
5x5	0.9033	0.9234	0.9133
7x7	0.9056	0.9039	0.9047

参考文献

[1] D. Hubel and T. Wiesel, “Receptive fields, binocular interaction and functional architecture in the visual cortex”, Journal of Physiology, pp.160:106–154, 1962.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation”, In Parallel Distributed Processing:Explorations in the Microstructures of Cognition, vol.1, pp.318–362. MIT Press, 1986.

[3] R. Vaillant, C. Monrocq and Y. LeCun, “Original approach for the localisation of objects in images”, IEE Proc on Vision, Image, and Signal Processing, No.141, Vol.4, pp.245–250, 1994.

[4] Y. LeCun, L. Bottou, Y. Bengio and P.Haffner, “Gradient-basedlearning applied to document recognition”, In Proceedings of the IEEE, Vol.86, No.11, pp. 2278–2324, 1998.

[5] P.Y. Simard, D. Steinkraus, and J.C. Platt, “Best practices for convolutional neural networks applied to visual document analysis”, In International Conference on Document Analysis and Recognition, Vol.2, pp. 958–962, 2003.

[6] S. Duffner and C. Garcia, “An online backpropagation algorithm with validation error-based adaptive learning rate”, In International Conference on Artificial Neural Networks (ICANN), Vol.1, pages 249–258, 2007.

[7] V. Jain, J.F. Murray, F. Roth, S. Turaga, V.Zhigulin, K.Briggman, M.Helmstaedter, W.Denk and H.S. Seung, “Supervised learning of

image restoration with convolutional networks", In IEEE International Conference on Computer Vision (ICCV2007), 2007.

[8] M. Delakisand and C. Garcia, "Text detection with convolutional neural networks", In InternationalConference on Computer Vision Theory and Applications (VISAPP 2008), 2008.

[9] D. Scherer, A. Muller, S.Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition", In International Conference on Artificial Neural Networks(ICANN2010), 2010.

[10] Y. Boureau, F. Bach, Y. LeCun and j. Ponce, "Learning Mid-Level Features For Recognition", In IEEE Conference on Computer Vision and Pattern Recognition(CVPR2010), 2010.

[11] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification", In IEEE Conference on Computer Vision and Pattern Recognition(CVPR2012), 2012.

[12] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional Neural Networks Applied to House Numbers Digit Classification", In International Conference on Pattern Recognition (ICPR 2012), 2012.

[13] A. Krizhevsky, I. Sutskever and G. Hinton, "Imagenet classification with deep convolutional neural networks", In Advances in Neural Information Processing Systems 25(NIPS2012), pp.1106-1114, 2012.

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors", arXiv preprint arXiv:1207.0580, 2012.

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

[16] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks", arXiv preprint arXiv:1311.2901, 2013.

[17] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. Lecun, "Pedestrian detection with unsupervised and multi-stage feature learning", In IEEE Conference on Computer Vision and Pattern Recognition(CVPR2013), 2013.

[18] W. Ouyang and X. Wang, "Single-pedestrian detection aided by multi-pedestrian detection", In IEEE Conference on Computer Vision and Pattern Recognition(CVPR2013), 2013.

[19] W.Ouyang and X. Wang "Joint Deep Learning for Pedestrian Detection", In IEEE International Conference on Computer Vision (ICCV2013), 2013.

[20] http://deeplearning.net/software/theano

[21] L. Wan, M. Zeiler, S. Zhang, Y. LeCun and R. Fergus, "Regularization of neural networks using dropconnect", In International Conference on Machine Learning (ICML13), 2013.

[22] P.Sermanet, D.Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks", arXiv preprint arXiv:1312.6229, 2013.