Reconhecimento de Caracteres em Imagens com Ruído

Sirlene Pio
UFOP
sirlenepg@gmail.com

Fernanda Maria Ribeiro
UFOP
fernandamaria_si@yahoo.com.br

Abstract

O desempenho dos métodos de aprendizagem de máquina depende geralmente da escolha da representação dos dados, extrair características de imagens de caracteres com ruído é uma tarefa difícil devido a baixa qualidade das imagens. Neste trabalho aplicamos uma rede neural convolucional (RNC) no problema de reconhecimento de caracteres com ruído. Os resultados obtidos na base de dados Street View House Numbers (SVHN) mostram que quanto maior o número de amostras de treinamento menor a taxa de erro, e que a não inclusão de camadas localmente conectadas na arquitetura da rede reduziu significativamente a acurácia. Na base de dados CAPCTHA CNPJ obtivemos $4,07\%$ *de erro, com uma arquitetura que além de incluir camadas localmente conectadas, também inclui camadas de normalização da resposta local.*

1. Introdução

O reconhecimento de padrões de imagens digitais tem sido cada vez mais estudado com o intuito de solucionar problemas cotidianos e auxiliar em algumas atividades realizadas por humanos, como reconhecimento biométrico, reconhecimento de objetos em geral e reconhecimento de caracteres com ruído. Esse último pode ser dividido em categorias: 1) reconhecimento de caracteres em imagens de cenas naturais e 2) reconhecimento de CAPTCHAs.

Sistemas de reconhecimento automático de imagens de cenas naturais podem ser aplicados em situações como no reconehecimento dos caracteres de placas de carros, pôsteres, letreiros, números de casas, etc. Os CAPTCHAs, teste de Turing público completamente automatizado para diferenciação entre computadores e humanos, são comumente utilizados para impedir que softwares automatizados executem ações que degradem a qualidade de algum serviço e também para proteger os sistemas vulneráveis ao spam. Pesquisas sobre sistemas de reconhencimento de CAPTCHAs, têm como objetivo prever ataques, mostrar as fragilidades de um certo modelo de CAPTCHA e indicar possíveis caminhos para o desenvolvimento de modelos de CAPTCHAs mais consistentes.

Representar de maneira robusta as características das imagens capturadas em ambientes reais e as informações discriminantes nos CAPTCHAs é uma questão em aberto no problema de reconhecimentode caracteres com ruído devido à baixa qualidade das imagens no primeiro, e aos ruídos e distorções presente no segundo. Imagens capturadas em cenas naturais geralmente possuem baixa qualidade, por serem adquiridas em diferentes perspectivas de iluminação, grau de oclusão e distância.

Neste trabalho utilizamos uma rede neural convolucional para extrair características e classificar imagens de caracteres com ruído. Realizamos experimentos em duas arquiteturas que se diferenciam apenas pela presença ou ausência de duas camadas localmante conectadas. O objetivo seria apurar a influência dessas camadas na rede convolucional. Para a primeira arquitetura, descrita na Subseção 4.3.3, realizamos experimentos variando o número de amostras de treinamento, o número de filtros da camada de convolução e a dimensão de tais filtros. Na base de dados CAPTCHA CNPJ obtivemos os melhores resultados aplicando uma terceira arquitetura que além de camadas localmente conectadas, também inclui camadas de normalização da resposta local.

A Seção 2 apresenta alguns trabalhos relacionados ao problema de reconhecimento de caracteres com ruído. A Seção 3 detalha os tipos de camadas suportadas pela rede neural utilizada nesse trabalho. Na Seção 4 apresentamos detalhes sobre as bases de dados utilizadas nos experimentos, detalhamos os mesmos e realizamos uma análise dos resultados obtidos. Finalmente a seção 5, expõe uma breve conclusão e possíveis trabalhos futuros.

2. Trabalhos Relacionados

Uma das metas principais em inteligência artificial é fazer com que a máquina aprenda a partir de exemplos (imagens, sons, dados) de um dado problema, de modo similar ao seres humanos. O aprendizado em profundidade busca este objetivo, com técnicas para aprender níveis de representação e abstração dos exemplos que sejam próximos dos seus significados.

O desempenho dos métodos de aprendizagem de máquina depende geralmente da escolha da representação dos dados (ou características) em que eles são aplicados, pois, a extração de características consiste em associar um vetor de características para cada imagem de modo que as imagens de uma mesma classe sejam representadas por vetores similares, ou seja, próximos no espaço de características.

Recentes resultados na neurociência forneceram conhecimentos sobre os princípios que regem a representação de informações no cérebro dos mamíferos, levando a novas ideias para projetar sistemas que representam informações. Uma das principais descobertas foi que o neocórtex, que está associado com muitas habilidades cognitivas, não explicitamente pré-processa os sinais sensoriais, mas sim, permite que se propaguem através de uma hierarquia complexa [8] de módulos que, ao longo do tempo, aprendem a representar observações com base nas regularidades que elas exibem [9]. Esta descoberta motivou o surgimento do subcampo de aprendizado profundo, que incide sobre modelos computacionais para obter representação de informações que apresentem características semelhantes ao do neocórtex [2].

Métodos de aprendizado profundo visam à representação de características de níveis mais altos da hierarquia formada pela composição de características de nível inferior [3], ou seja, esses métodos consistem em transformar a representação do pixel bruto em representações gradativamente mais abstratas.

Ao longo dos últimos anos vários estudos demonstraram a eficácia dos métodos de aprendizado profundo em muitos domínios de aplicações. Além do MNIST [1] desafio da escrita, existem aplicações na detecção de face [6] [11], reconhecimento e detecção de fala [14], reconhecimento de objetos em geral e processamento da linguagem natural [7].

Os descritores de apredizagem em profundidade são aplicados ao reconhecimento de caracteres em imagens com ruído nos trabalhos [10], [13] e [5]. Em [10] os autores mostram as principais vantagens de dois descritores de aprendizagem profunda: 1) uma rede neural convolucional que treina os filtros utilizando uma variante do algoritmo K-means (Spherical K-means) e não utiliza o algoritmo Back-propagation para treinar a rede [4] (obteve melhor acurácia na base SVHN, $90,6\%$) e 2) Uma rede neural que se baseia no empilhamento de *auto-encoder* esparsos, em relação aos extratores de característica *hand-designed*: 1) *Histograms-of-Oriented-Gradients* (HOG) e 2) *Binary Features* (WDCH). Já em [13] os autores incluiram na arquitetura da rede neural convolucional tradicional a abordagem de aprendizado de características multi-estágio usando *Lp pooling* e alcançam uma taxa de precisão igual a $95,1\%$ na base de dados SVHN. Mais recentemente, os autores em [5] propõe uma abordagem unificada que integra os três passos das abordagens tradicionais de reconhecimento de imagens com ruído, localização, segmentação e reconhecimento, através da utilização de um neural convolucional profunda rede que opera diretamente com os pixels da imagem, e alcançam $97,84\%$ de precisão na base SVHN.

3. Metodologia

Neste trabalho a rede convolucional implementada em C++/CUDA por Alex Krizhevsky[1] foi aplicada ao problema de reconhecimento de caracteres com ruído. CUDA (Arquitetura de Computação de Dispositivos Unificados) é uma plataforma de computação paralela e um modelo de programação que permite aumentos significativos de performance computacional ao aproveitar a potência da unidade de processamento gráfico (GPU).

A rede neural convolucional, em questão, é treinada pelo algoritmo Back-propagation. O algoritmo de aprendizagem Back-propagation é um algoritmo supervisionado, também conhecido como Regra Delta Generalizada, e baseia-se no método gradiente descendente. O procedimento requer duas fases: propagação da ativação, e retropropagação do erro. Primeiramente, propaga-se o erro nas camadas que não compartilham pesos e posteriormente nas camadas que compartilham pesos.

3.1. Camadas da Rede Neural

3.1.1 Camada de Convolução

A invariância à translação pode ser obtida em redes neurais artificiais através de uma técnica chamada compartilhamento de pesos. O compartilhamento de peso é uma restrição que obriga determinados pesos de conexões entre neurônios a possuir exatamente o mesmo valor. Desta forma, os neurônios deste conjunto podem responder de forma igual quando um padrão é apresentado a um ou a outro neurônio. A técnica de compartilhamento de pesos pode ser vista como uma operação de convolução [12].

Convolução é o processo de calcular a intensidade de um determinado pixel em função da intensidade de seus vizinhos, a definição de convolução é dada pela Equação 1. Uma forma de interpretar esta equação é considerar cada pixel da imagem de saída, $I(x,y)^* D_i(x,y)$, como sendo uma soma ponderada de pixels próximos na imagem de entrada, I, onde os pesos e o tamanho da região onde esta média é calculada são definidos pelo núcleo da convolução f.

$$I(x,y)^* f(x,y) = \sum_s \sum_t I(s,t) D_i(x-s, y-t) \quad (1)$$

onde (x,y) são as coordenadas da origem do núcleo. (s,t) deve cobrir todo o núcleo f.

[1] https://code.google.com/p/cuda-convnet/

O resultado da convolução é então submetido a função de ativação, que neste trabalho é dada por:

$$f(x) = max(0, x) \tag{2}$$

3.1.2 Camada *Max-pooling*

A saída da camada *max-pooling* é dada pela máxima ativação de regiões retangulares sem sobreposição. Essa camada reduz a resolução do mapa de características. Tem o efeito de reduzir a sensibilidade da saída do mapa a deslocamentos e outras formas de distorção, selecionando características invariantes que melhoram o desempenho de generalização.

3.1.3 Camada de Normalização da Resposta Local (mesmo mapa)

Este tipo de camada calcula a função

$$f(u_f^{x,y}) = \frac{u_f^{x,y}}{(1 + \frac{\alpha}{N^2} \sum_{x'=max(0,x-\lfloor N/2 \rfloor)}^{min(S,x-\lfloor N/2 \rfloor+N)} \sum_{y'=max(0,y-\lfloor N/2 \rfloor)}^{min(S,y-\lfloor N/2 \rfloor+N)} (u_f^{x',y'})^2)^\beta} \tag{3}$$

onde $u_f^{x,y}$ representa a atividade de uma unidade do mapa f na posição x, y antes da normalização, S é o tamanho da imagem, e N é o tamanho da região a ser usado para a normalização.

Essa camada permite a detecção de características de alta frequência, com uma grande resposta dos neurônios, enquanto amortece respostas que são uniformemente grandes em uma vizinhaça local.

3.1.4 Camada Localmente Conectada

A camada localmente conectada tem as mesmas funcionalidades da camada convolucional, mas, esta, não compartilha pesos.

3.1.5 Camada Totalmente Conectada

A camada totalmente conectada pode ser ilustrada por um grafo bipartido, onde todos os neurônios de um nível têm ligação com todos os neurônios do nível seguinte.

As camadas totalmente conectada, softmax e regressão logística constituem a etapa de classificação neste trabalho. O número de neurônios de saída da camadas totalmente conectada equivale ao número de classes do problema, que neste caso pode ser 10 para a classe SVHN e 36 para a base CAPTCHA CNPJ, como descrito na Seção 4.

3.1.6 Camada Softmax

A Equação 4, onde x é o vetor de entrada, apresenta uma transformação exponencial normalizada. A camada softmax produz estimativas positivas ($f(x_i) >= 0, \forall i$) cuja soma é igual a um, que são interpretadas como probabilidades.

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{4}$$

3.1.7 Camada Regressão Logística

A rede implementada por Alex Krizhevsky define a regressão logística multinomial como o objetivo a ser otimizado. Um modelo de regressão pode ser definido como uma equação matemática em que se expressa o relacionamento de variáveis. Nestes modelos, define-se uma variável dependente, neste caso as classes possíveis, ou variável de saída, e procura-se verificar a influência de uma ou mais variáveis ditas variáveis independentes, no caso as probabilidades computadas na camada softmax, sobre esta variável dependente.

4. Experimentos

4.1. Base de Dados

4.1.1 SVHN

A base de dados SVHN[2] foi obtida a partir de números de casa de imagens no Google do Street View. A base de dados contém mais de 600 mil imagens digitais, sendo 73.257 dígitos para treinamento, 26.032 dígitos para testes, e 531.131 amostras adicionais, mais simplificadas que podem ser utilizadas como para complementar a conjunto de treinamento. Os caracteres dessa base são exclusivamente números.

A SVHN é disponibilizado em dois formatos:

1. Formato 1: imagens originais com caixas delimitadoras de resolução variável.

2. Formato 2 (*cropped*): As caixas delimitadoras dos caracteres são estendidos na dimensão adequada para se tornarem janelas quadradas, para que sejam redimensionandas para uma dimensão fixa, 32-por-32 pixels. Esse redimensionamento não introduz distorções de *aspect ratio*, porém introduz dígitos (ou partes) além do dígito de interesse.

   Nos experimentos deste trabalho utilizamos apenas o formato *cropped*. As Figuras 1 e 2 apresentam exemplos de figuras no formato 1 e 2, respectivamente.

[2] http://ufldl.stanford.edu/housenumbers/

Figura 1. SVHN - Formato 1.

Figura 2. SVHN - Formato 2.

4.1.2 CAPTCHA CNPJ

A base de dados CAPTCHA CNPJ[3] é composta de 12 mil CAPTCHAs, sendo que cada um deles é constituído por seis caracteres. Neste trabalho, a base foi dividida em um conjunto de treinamente (60 mil caracteres) e conjunto de teste (12 mil caracteres). Esta base possui 36 classes de caracteres possíveis, 10 números $[0-9]$ e 26 letras $[A-Z]$. Os caracteres, que constituem os CAPTCHAs dessa base, possuem distorções, tamanhos variados, sobreposição, e muitas vezes estão incompletos. Os CAPTCHAs também apresentam ruídos, como pontos, linhas e curvas, na mesma cor dos caracteres, dois exemplos de imagem dessa base são ilustrados na Figura 3.

Figura 3. Exemplos de CAPTCHA da base CAPTCHA CNPJ.

4.1.3 Configuração dos Dados de Entrada para a Rede Convolucional

Os dados são divididos em lotes. O número mínimo de lotes é três: 1) um para treinamento, 2) um para validação e 3) um para teste. Cada lote é composto por uma matriz que armazena os dados e um vetor referente aos rótulos ou classes. Cada linha da matriz de dados representa uma amostra, por exemplo, se cada amostra possui dimensões 32-por-32 pixels e três canais (RGB), a matriz de dados terá 3072 colunas. O vetor de rótulos possui dimensão igual a 1-por-(número de amostras).

[3] rp20142@lapdiftp.decom.ufop.br

4.2. Detalhes do Treinamento

O procedimento para treinar a rede convolucional utilizada neste trabalho é dividido em quatro etapas:

1. A rede é inicialmente treinada sobre os lotes 1 a $(n-1)$, sendo n o número total de lotes de treinamento, e o teste é realizado sobre o lote n. O treinamento é realizado até que o erro de validação pare de melhorar.

2. Na segunda etapa a rede é treinada sobre todos os lotes de treinamento, 1 a n. O treinamento é realizado até que o erro de treinamento no lote n (que costumava ser o nosso conjunto de validação), seja o mesmo obtido na etapa anterior.

3. Todas as taxas de aprendizagem são reduzidas por um fator de 10, e treina-se por mais 10 épocas. Quanto menor for a taxa de aprendizado $epsW$, menores as mudanças nos pesos sinápticos entre duas iterações e mais suave a trajetória no espaço de pesos.

4. Todas as taxas de aprendizagem são reduzidas por um novo fator de 10, e treina-se por mais 10 épocas.

4.3. Arquiteturas

4.3.1 Arquitetura Conv-Local

A arquitetura Conv-Local é ilustrada na Figura 4, ela é formada por duas sequências de uma camada de convolução e uma de *pooling*, seguidas por duas camadas localmente conectadas, uma camada completamente conectada, uma camada softmax e uma camada que realiza a regressão logística.

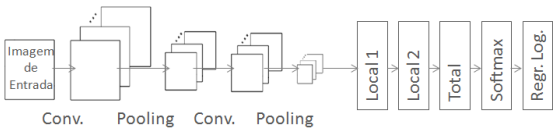

Figura 4. Arquitetura Conv-Local

A Tabela 1 apresentra os resultados dos experimentos realizados na base de dados SVHN. Foram realizados experimentos variando-se o número de amostras de treinamento, o número de filtros das camadas de convolução, assim como suas dimensões. Para todas quatro possibilidades de número de amostra de treinamento $[70k, 150k, 300k, 600k]$

a melhor taxa de erro foi obtida utilizando 128 filtros de dimensões 7-por-7 e a pior adotando 128 filtros de dimensões 3-por-3. Esse resultado indica que há correlação entre esses dois parâmetros número de filtros e dimensões dos mesmos.

Tabela 1. Resultados obtidos na base de dados SVHN (Arquitetura Conv-Local)

SVHN			
# amostras de treinamento	# filtros	Tamanho dos filtros	Taxa de erro (%)
70k	32	5	24,95
70k	64	5	22,83
70k	128	5	22,35
70k	128	3	25,72
70k	128	7	**20,25**
150k	32	5	18,52
150k	64	5	16,49
150k	128	5	15,54
150k	128	3	18,38
150k	128	7	**14,93**
300k	32	5	14,36
300k	64	5	13,04
300k	128	5	12,91
300k	128	3	14,85
300k	128	7	**11,65**
600k	32	5	11,46
600k	64	5	10,28
600k	128	5	9,97
600k	128	3	11,45
600k	128	7	**9,35**

Realizamos experimentos na base de dados CAPTCHA CNPJ na arquitetura Conv-Local, utilizando os parâmetros que obtiveram a menor taxa de erro na base SVHN, 128 filtros de dimensões 7-por-7. A taxa de erro obtida é apresentada na Tabela 2.

Tabela 2. Resultados obtidos na base de dados CAPTCHA CNPJ (Arquitetura Conv-Local)

CAPTCHA CNPJ			
# amostras de treinamento	# filtros	Tamanho dos filtros	Taxa de erro (%)
10k	128	7	5,94

4.3.2 Arquitetura Conv

A arquitetura Conv é formada por duas sequências de uma camada de convolução e uma de *pooling*, seguidas por uma camada completamente conectada, uma camada softmax e uma camada que realiza a regressão logística. A arquitetura Conv é apresentada na Figura 5

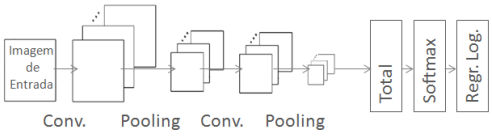

Figura 5. Arquitetura Conv

Selecionamos os parâmetros que obtiveram a menor taxa de erro para a arquitetura Conv-Local, e testamos nessa arquitetura de RNC sem camadas localmente conectadas. Realizamos este experimento com o intuito de verificar e quantificar a influência da presença ou ausência das camadas localmente conectadas na rede neural de convolução. Na Tabela 3 observamos que a acurácia obtida nessa arquitetura na base de dados SVHN foi $4,14\%$ inferior a obtida na arquitetura Conv-Local. Porém, para a base de dados CAPTCHA CNPJ a acurácia obtida aqui foi $1,36\%$ superior a obtida na arquitetura Conv-Local.

Tabela 3. Resultados obtidos nas bases de dados SVHN e CAPTCHA CNPJ (Arquitetura Conv)

Base de dados	# amostras de treinamento	# filtros	Tamanho dos filtros	Taxa de erro (%)
SVHN	600k	128	7	13,49
CAPTCHA CNPJ	10k	128	7	4,58

4.3.3 Arquitetura Conv-Norm-Local

A arquitetura Conv-Norm-Local é ilustrada na Figura 4, como podemos observar essa arquitetura é formada por duas sequências de uma camada de convolução, uma de *pooling* e uma de normalização da resposta local (mesmo mapa), seguidas por duas camadas localmente conectadas, uma camada completamente conectada, uma camada softmax e uma camada que realiza a regressão logística.

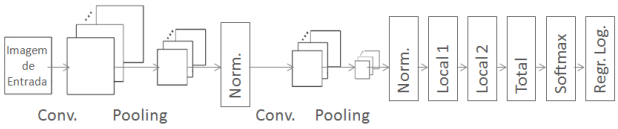

Figura 6. Arquitetura Conv-Norm-Local

A Tabela 4 apresenta os resultados obtidos nessa arquitetura para a base de dados CAPTCHA CNPJ. Apoiados nos resultados obtidos na Tabela 1, realizamos experimentos aumentando gradativamente o número de filtros da camada de convolução, na expectativa de que a taxa de erro diminuísse. Porém, a menor taxa de erro, $4,07\%$, novamente foi obtida adotando 128 filtros de dimensões 7-por-7 na camada de convolução.

Tabela 4. Resultados obtidos nas bases de dados CAPTCHA CNPJ (Arquitetura Conv-Norm-Local)

CAPTCHA CNPJ			
# amostras de treinamento	# filtros	Tamanho dos filtros	Taxa de erro (%)
10k	64	7	4,42
10k	128	7	**4,07**
10k	192	7	4,33

4.4. Análise dos Resultados

A melhor taxa de erro na base SVHN foi obtida utilizando 128 filtros de dimensões 7-por-7 e a pior adotando 128 filtros de dimensões 3-por-3. Esse resultado indica que

há correlação entre esses dois parâmetros número de filtros e dimensões dos mesmos. A Figura 7 apresenta a relação entre a taxa de erro e o número de amostras de treinamento, podemos observar que quanto maior o número de amostras de treinamento menor é a taxa de erro.

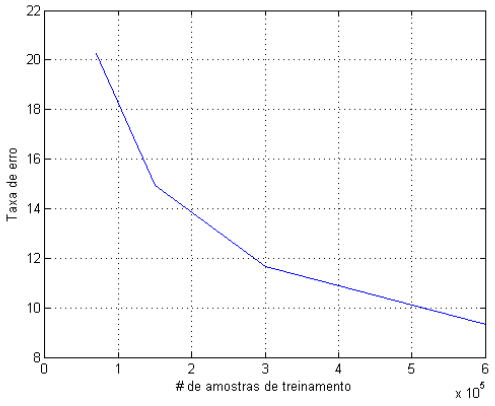

Figura 7. Taxa de erro vs número de amostras de treinamento.

Para a base CAPTCHA CNPJ o melhor resultado foi obtido na arquitetura Conv-Norm-Local, também adotando 128 filtros de dimensões 7-por-7 na camada de convolução.

5. Conclusão e Trabalhos Futuros

Neste trabalho utilizamos uma rede neural convolucional para extrair características e classificar imagens de caracteres com ruído. Realizamos testes em duas arquiteturas de rede neural convolucional na base SVHN, com e sem camadas localmente conectadas, sendo que essa última obteve acurácia $4,14\%$ inferior em relação a primeira. Para a base SVHN, também realizamos experimentos na arquitetura com camadas localmente conectadas variando o número de amostras de treinamento, o número e as dimensões do filtro de treinamento. Analisando os resultados concluímos que para essa arquitetura, quanto maior o número de amostras menor é a taxa de erro, e que o número de filtros e a dimensão dos mesmos são parâmetros correlacionados.

Já na base CAPTCHA CNPJ, realizamos experimentos nas duas arquiteturas citadas e em uma terceira que incluía camadas de normalização da resposta local, essa última arquitetura obteve os melhores resultados. Diferentemente dos resultados obtidos na base SVHN, na base CAPTCHA CNPJ, a arquitetura sem camadas localmente conectadas obteve uma taxa de erro menor que a arquitetura com tais camadas.

Futuramente, apoiados ao gráfico ilustrado na Figura 7 que indica uma redução gradativa na taxa de erro à medida que o conjunto de treinamento cresce, aumentaremos o número de amostras de treinamento gerando imagens sintéticas, através da aplicação de rotações, espelhamento, inserção de ruídos, etc, nas imagens originais.

Referências

[1] The mnist database of handwritten digits [online]. http://yann.lecun.com/exdb/mnist/.

[2] I. Arel, D. C. Rose, and T. P. Karnowski. Deep machine learning: A new frontier in artificial intelligence research. *IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE*, 2010.

[3] Y. Bengio. Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, 2009.

[4] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. NG. Text detection and character recognition in scene images with unsupervised feature learning. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 440–445. IEEE, 2011.

[5] J. I. S. A. V. S. Ian J. Goodfellow, Yaroslav Bulatov. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv:1312.6082v4*, 2014.

[6] B. Kwolek. Face detection using convolutional neural networks and gabor filters. *Lecture Notes in Computer Science*, 2005.

[7] H. Lee, Y. Largman, P. Pham, , and A. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. *Advances in Neural Information Processing Systems 22 (NIPS'09)*, 2009.

[8] T. Lee and D. Mumford. Hierarchical bayesian inferencein the visual cortex. *J. Opt. Soc. Amer.*, 2003.

[9] T. Lee, D. Mumford, R. Romero, and V. Lamme. The role of the primary visual cortex in higher level vision. *Vision Res.*, 1998.

[10] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4, 2011.

[11] M. Osadchy, Y. LeCun, and M. Miller. Synergistic face detection and pose estimation with energy-based models. *J. Mach. Learn. Res.*, 2007.

[12] J. M. Otuyama. *Rede Neural por Convolução para Reconstrução Estéreo*. PhD thesis, Universidade Federal de Santa Catarina, 2000. Dissertação de Mestrado.

[13] S. C. Pierre Sermanet and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. In *International Conference on Pattern Recognition*, 2012.

[14] S. Sukittanon, A. C. Surendran, J. C. Platt, and C. J. C. Burges. Convolutional networks for speech detection. *Interspeech*, 2004.