3D Objects Recognition Using Artificial Neural Networks

Diogo Santos Ortiz Correa
Laboratório de Robótica Móvel
Universidade de São Paulo
São Carlos, Brasil
correa.dso@gmail.com

Fernando Santos Osório
Laboratório de Robótica Móvel
Universidade de São Paulo
São Carlos, Brasil
fosorio@icmc.usp.br

***Resumo*—The recent advances in computational processing and the low prices of sensors able to capture three-dimensional information have contributed for the progress of computer vision researches involving 3D data and 3D images. Object recognition allows us to develop complex applications for intelligent mobile robotics, augmented reality, systems for the visually impaired, among other applications. In this context, this paper presents a method for recognizing and classifying objects which are represented in three dimensions through depth maps. The data used in this study comes from the "UW RGB-D Object Dataset" from University of Washington, which is available online and is largely used to evaluate 3D object classifiers. This object database is composed of depth maps captured by the Microsoft's Kinect sensor. The obtained results are promising and contribute positively to the computer vision area.**

***Index Terms*—Object recognition, Three-dimensional images, Artificial neural networks, Kinect**

I. Introdução

Nos últimos anos tem aumentado os esforços em pesquisas relacionadas ao reconhecimento 3D de objetos. Isto se dá devido ao aumento de poder computacional dos computadores e ao barateamento dos sensores de captura de dados tridimensionais. Além disso, há uma grande aplicabilidade destas tecnologias em áreas como a saúde [1], a realidade aumentada, a robótica inteligente [2], a interação humano-robô, e inclusive na educação [3].

O objetivo deste trabalho é apresentar o desenvolvimento de um classificador de objetos 3D, usando Redes Neurais Artificiais para reconhecer objetos que estão representados tridimensionalmente (i.e., imagens com dados de profundidade) obtidas a partir do sensor Kinect. Atualmente muito tem sido pesquisado e desenvolvido visando o reconhecimento de imagens tradicionais 2D, com uso de Redes Neurais Artificiais Tradicionais e Redes Neurais Profundas e Convolucionais (*Deep Learning* & CNN), porém poucos trabalhos abordam o reconhecimento de objetos com dados 3D.

Redes Neurais Artificiais (RNAs), sejam elas rasas (*Shallow* NN) ou profundas (*Deep* NN), são modelos matemáticos que se assemelham e/ou se inspiram nas estruturas neurais biológicas e que têm capacidade computacional adquirida por meio de adaptação, apresentando propriedades de aprendizado a partir de exemplos e de generalização [4]. As Redes Neurais podem ser capazes de reconhecer objetos em imagens, a exemplo dos destacados resultados obtidos com as RNAs em competições como o ImageNet *Large Scale Visual Recognition Competition* (ILSVRC)[1] e no COCO *Large-scale object detection, segmentation, and captioning dataset*[2]. Redes Neurais como o YOLO[3] são capazes de reconhecer um grande número de objetos, porém somente usando imagens tradicionais, não sendo capazes de diferenciar uma foto (imagem plana 2D) de uma cena real com profundidade (3D).

O Kinect é um dispositivo desenvolvido principalmente para uso em jogos eletrônicos do console XBox 360/One, da Microsoft, lançado inicialmente em novembro de 2010. O Kinect é um dispositivo RGB-D, ou seja, permite a captura de imagens representando as cores (RGB) e a profundidade de uma cena (D – *Depth*). Portanto, uma cena capturada pelo Kinect é usualmente representada por um par de imagens de resolução aproximada de 640x480, com uma imagem em formato colorido RGB (24 bits/pixel) e a outra imagem representando a profundidade de cada pixel (*Depth*), também chamada de mapa de profundidade. A profundidade representa a distância dos pixels em relação ao sensor, formando um mapa LxCxP (Linha x Coluna x Profundidade). A percepção de ambientes usando dispositivos 3D (e.g. Kinect, Câmeras Estéreo) é cada vez mais usada, pois afinal, vivemos e interagimos com objetos em um mundo tridimensional.

II. Trabalhos relacionados

As pesquisas em reconhecimento de objetos em imagens RGB e RGB-D tem avançado nos últimos anos. Em [5], é realizada a fusão de RGB e D, ou seja, realiza a classificação utilizando tanto imagens coloridas quanto o mapa de profundidade.

Nos trabalhos de [6], [7], [8], [9] e [10] foram desenvolvidos descritores de características para representação de nuvens de pontos tridimensionais. Apresentam invariância a posição, orientação e escala. Nestes trabalhos foi realizada também a classificação de objetos utilizando Redes Neurais e os descritores desenvolvidos.

Os autores agradecem à CAPES e ao CNPq pelo apoio financeiro durante este trabalho.

[1]IMAGENET ILSVRC (2D)- http://www.image-net.org/challenges/LSVRC
[2]COCO Dataset (2D) - http://cocodataset.org/#home
[3]Darknet YOLO - https://pjreddie.com/darknet/yolo/

Em [11] são combinadas redes neurais profundas com classificação gaussiana. Utiliza-se uma parte dos dados rotulados manualmente e uma outra grande parte dos dados sem rótulos para realizar a classificação. Realiza o processo de classificação em tempo real. Ver Figura 1.

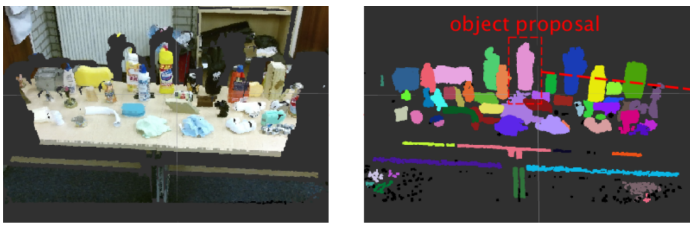


Figura 1. Classificação de objetos usando nuvens de pontos. À esquerda nuvem de pontos e à direita objetos classificados. Fonte: [11]

Em outro trabalho foram realizados três tipos de classificação ao mesmo tempo: reconhecimento de categoria (qual o tipo de objeto), reconhecimento de instância (alguma característica do objeto) e reconhecimento de pose (posição do objeto no ambiente) [12]. Primeiro é classificada a categoria para saber qual o tipo de objeto é aquele. Depois é realizado o reconhecimento da instância para saber, por exemplo, a marca de um objeto. E, por fim, é classificada a pose, para saber se o objeto está apontado para direita ou para esquerda, por exemplo. Ver Figura 2.

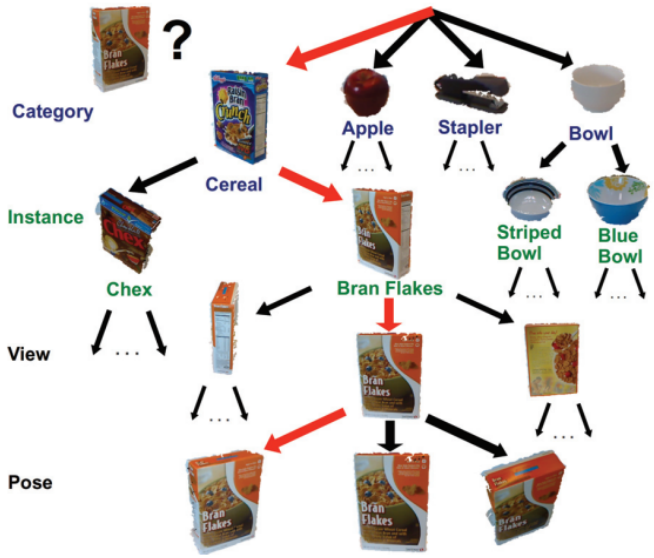


Figura 2. Reconhecimento de categoria, instância e pose. Fonte: [12]

No trabalho de [13] são utilizadas Redes Neurais Convolucionais (Aprendizado profundo - CNN) para aprendizagem e reconhecimento de objetos 3D. Utiliza uma base de dados em que os objetos estão armazenados em um formato tridimensional chamado *mesh*. Primeiramente transforma para nuvem de pontos e então realiza a classificação.

Em [3] é apresentado um classificador de objetos incorporado em um robô humanoide com aplicação na educação. Este robô é dotado de sensores e identifica objetos apresentados a ele. Feito o processamento, exibe comportamentos distintos de acordo com o resultado.

Estes trabalhos apresentam técnicas de extração de atributos e reconhecimentos de objetos 3D, porém, em sua maioria são técnicas que exigem dados representados em formatos mais complexos (e.g. *mesh*, *point clouds*, *octrees*), sendo necessário, no caso de bases de dados RGB-D (Kinect), a conversão para estruturas de dados auxiliares. Além disto, tais métodos possuem também algumas restrições impostas quanto a performance, inclusive devido as transformações e processamentos mais complexos dos dados que são necessários.

III. Desenvolvimento e resultados

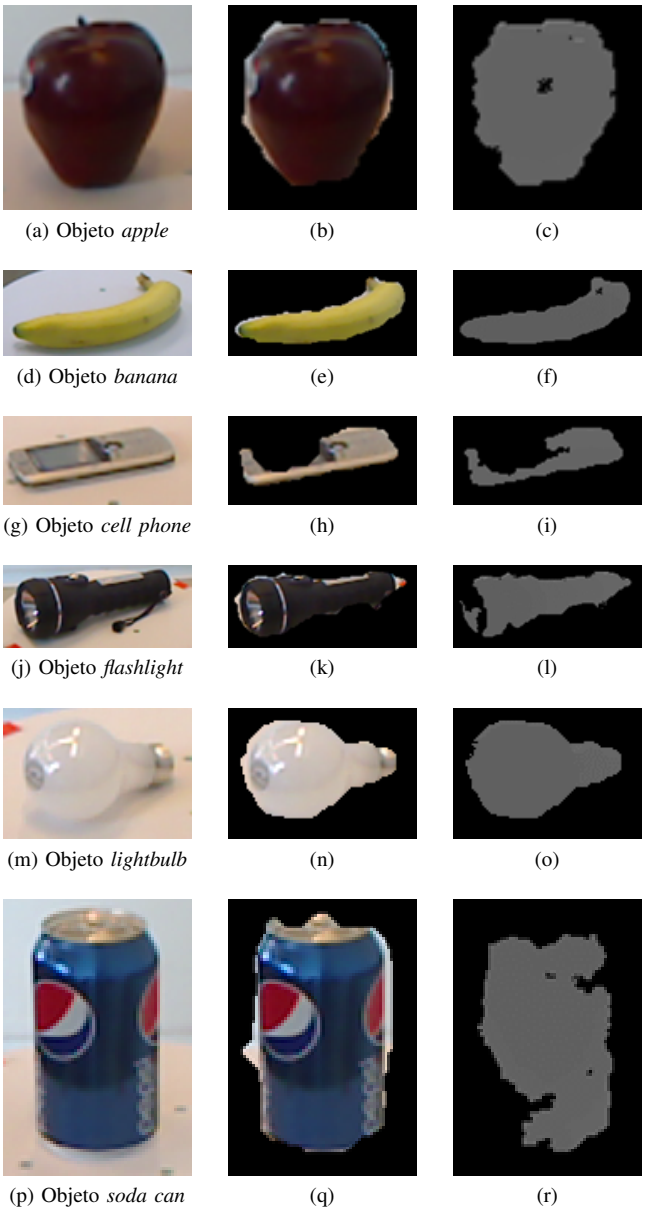
Escolheu-se a base de dados *UW - "RGB-D Object Dataset"* da Universidade de Washington [14], que contém imagens coloridas (RGB) e mapas de profundidade (D) obtidas com o sensor Kinect. Esta base é amplamente utilizada em pesquisas de reconhecimento de objetos 3D. Esta base contém imagens de objetos comuns em ambiente doméstico e escritório. Para realizar a captura, os objetos foram colocados sobre uma base giratória. Assim, foi possível capturar imagens com várias poses diferentes dos objetos. O Kinect foi disposto a 1 metro de distância do objeto e foram realizadas capturas em 3 pontos de vista diferentes (posicionamento do Kinect em relação ao objeto): 30, 45 e 60 graus a partir do horizonte. Com isto o objeto é rotacionado em diferentes eixos: ao redor próprio eixo e inclinado em relação ao Kinect. Os autores da base realizaram a segmentação dos objetos. A segmentação de objetos 3D pode ser mais facilmente realizada do que em imagens 2D, pois a profundidade do objeto é usada para separar o mesmo dos elementos de fundo ou mesmo da superfície de apoio. A base do *UW RGB-D* fornece os objetos já previamente segmentados.

Os objetos podem ser separados em 51 categorias diferentes. Destes objetos foram selecionadas 6 categorias (Figura 3) para extrair atributos, treinar e classificar as Redes Neurais Artificiais. Os objetos que compõem estas 6 categorias estão armazenados em 17.581 imagens coloridas (RGB) e o mesmo número de imagens de mapa de profundidade. As imagens coloridas não foram utilizadas neste trabalho. Apenas são exibidas na Figura 3 para orientar o leitor com relação aos objetos utilizados.

Foram separadas para o treinamento 11588 imagens de profundidade, o que representa aproximadamente 2/3 das imagens. Para a classificação foram separadas 5993 imagens de profundidade, o que representa aproximadamente 1/3 das imagens de profundidade.

A. *Pré-processamento*

A biblioteca utilizada para ler e processar as imagens foi a OpenCV [15]. A base de dados utilizada fornece as imagens dos objetos e uma máscara de segmentação. Após utilizar esta máscara foi obtido o resultado, conforme a Figura 4. Ao aplicar a máscara, o fundo é removido da imagem restando apenas os objetos.



(a) Objeto *apple* (b) (c)

(d) Objeto *banana* (e) (f)

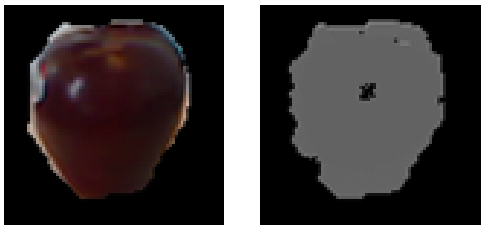
(g) Objeto *cell phone* (h) (i)

(j) Objeto *flashlight* (k) (l)

(m) Objeto *lightbulb* (n) (o)

(p) Objeto *soda can* (q) (r)

Figura 3. Categorias de objetos escolhidos para realização deste trabalho. À esquerda imagens RGB, no centro imagens RGB segmentadas e à direita mapas de profundidade segmentados. Fonte: Elaborada pelo autor.



(a) Imagem RGB (b) Mapa de profundidade

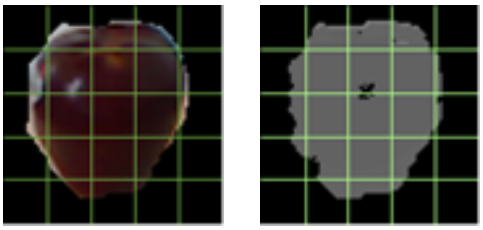
Figura 4. Imagens segmentadas utilizando máscara. Fonte: Elaborada pelo autor.

Antes de fazer o treinamento da Rede Neural é necessário formar vetores de características para alimentar o algoritmo da Rede Neural. Cada vetor representa em números cada imagem de profundidade. Para formar um vetor fez-se a extração de características da imagem de profundidade, portanto foram usadas apenas as informações de profundidade (*depth*) da imagem, não sendo consideradas na classificação as informações de cor e textura (RGB). Foram desenvolvidos alguns algoritmos para extrair estes atributos, conforme apresenta a Tabela I.

Tabela I
ATRIBUTOS EXTRAÍDOS DAS IMAGENS. FONTE: ELABORADA PELO AUTOR.

Atributo	Valores no vetor de características
densidades	compõe 25 valores no vetor de características
altura	compõe 1 valor no vetor de características
largura	compõe 1 valor no vetor de características
soma linhas	compõe 5 valores no vetor de características
soma colunas	compõe 5 valores no vetor de características

1) Atributo *densidades*: O mapa de profundidade foi dividido em 25 regiões iguais. Em cada região contou-se o número de pixels do objeto que ocupa aquela região. Ver Figura 5.



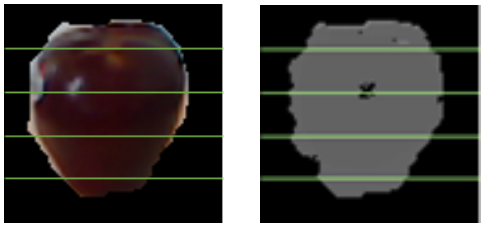
(a) Imagem RGB (b) Mapa de profundidade

Figura 5. Imagens segmentadas. As linhas verdes apresentam as divisões das regiões. Fonte: Elaborada pelo autor.

2) Atributo *altura*: Para obter este valor, calculou-se a altura em pixels do objeto na imagem, ou seja, a distância entre o pixel inferior até o pixel superior do objeto.

3) Atributo *largura*: Para obter este valor, calculou-se a largura em pixels do objeto na imagem, ou seja, a distância entre o pixel mais à esquerda e o pixel mais à direita do objeto na imagem.

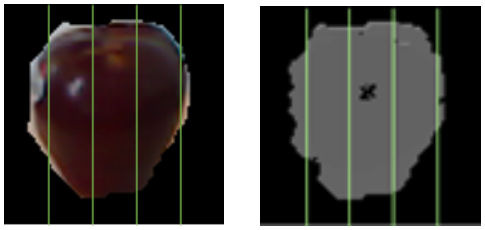
4) Atributo *soma linhas*: A imagem do objeto pode ser dividida em 5 retângulos horizontais. Cada um destes retângulos é considerado uma região (Figura 6). Para cada uma destas 5 regiões contou-se o número de pixels do objeto presente. Um outro algoritmo utilizado para somar as linhas é contar o valor dos pixels dos objetos em cada região. Soma-se o valor do pixel, ou seja, considera a profundidade que aquele pixel representa. O primeiro algoritmo foi chamado de *binário* e o segundo de *z-pixel*.



(a) Imagem RGB (b) Mapa de profundidade

Figura 6. Imagens segmentadas. As linhas verdes apresentam as divisões das regiões. Fonte: Elaborada pelo autor.

5) Atributo *soma colunas*: Semelhante ao atributo *soma linhas*, porém, separam-se as regiões em retângulos verticais, formando 5 colunas (Figura 7). Da mesma forma, podem ser somadas utilizando o algoritmo *binário* (somam-se o número de pixels) ou o *z-pixel* (somam-se os valores dos pixels).



(a) Imagem RGB (b) Mapa de profundidade

Figura 7. Imagens segmentadas. As linhas verdes apresentam as divisões das regiões. Fonte: Elaborado pelo autor.

Todos os valores dos atributos são normalizados com valores entre 0 e 1. Na normalização do atributo *densidades* é considerado apenas os valores da própria imagem. Na normalização das outras entradas (*altura*, *largura*, *soma linhas* e *soma colunas*) está considerando todas as imagens do treinamento (quando faz o treinamento) ou todas as imagens da classificação (quando faz a classificação). Cabe destacar que, a classificação é feita em lote como o treinamento.

O vetor de características pode ser formado apenas pelos valores do atributo *densidades* ou pode ser combinado com os atributos *altura*, *largura*, *soma linhas* e *soma colunas*. Quando *soma linhas* e *soma colunas* são utilizados, ambos utilizam apenas uma forma de contagem: *binário* ou *z-pixel*.

Após a extração dos atributos e inicialização do vetor de características, um arquivo é salvo e a Rede Neural já pode ser treinada. Nestes arquivos também são salvas as saídas desejadas para cada imagem.

B. Treinamento

É utilizado um arquivo de configuração em que são indicados os arquivos que contém os dados para o treinamento, bem como parâmetros da rede neural.

A topologia da rede neural artificial (Figura 8) é formada por três camadas: entrada, intermediária e saída. A camada de entrada varia entre 25 e 37 neurônios, de acordo com os atributos utilizados. 25 (*densidades*) + 5 (*soma colunas*) + 5 (*soma linhas*) + 1 (*largura*) + 1 (*altura*). Na camada intermediária foi utilizado o mesmo número de neurônios que a camada de entrada. A camada de saída sempre tem 6 neurônios que é o número de saídas desejadas, ou seja, o número de categorias.

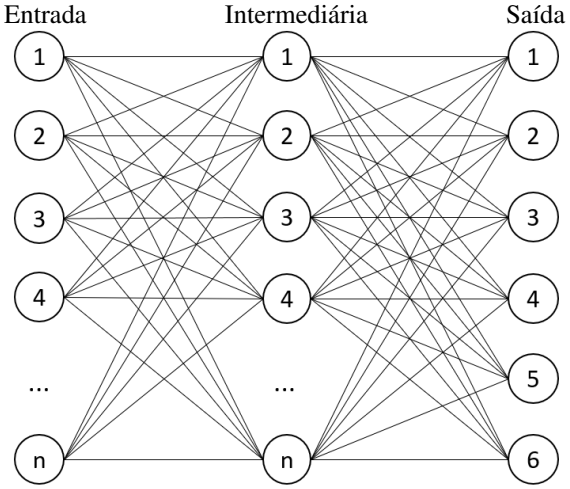


Figura 8. Topologia da Rede Neural. O valor de *n* pode variar entre 25 e 37. Fonte: Elaborada pelo autor.

É utilizada a codificação *one-hot* das classes dos objetos, conforme apresentado na Tabela II. Nesta codificação cada classe de objeto é representado por um vetor diferente.

Tabela II
CODIFICAÇÃO ONE-HOT DAS CLASSES DOS OBJETOS. FONTE: ELABORADA PELO AUTOR.

apple	1	0	0	0	0	0
banana	0	1	0	0	0	0
cell phone	0	0	1	0	0	0
flashlight	0	0	0	1	0	0
lightbulb	0	0	0	0	1	0
soda can	0	0	0	0	0	1

Para realizar o treinamento da Rede Neural foi utilizada a biblioteca FANN (*Fast Artificial Neural Networks*) [16]. O algoritmo de treinamento usado foi o RPROP (*Resilient Propagation*). A taxa de aprendizado utilizada foi 0.7. O número de épocas de treinamento foi 10 mil. Um arquivo contendo a Rede Neural bem como os pesos de cada neurônio é o resultado do treinamento.

C. Classificação

No início do processo de classificação é lido um arquivo de configuração que contém o endereço do arquivo contendo a rede neural e os endereços das pastas que contém as imagens para classificar.

Então são lidas as imagens que serão classificadas. Destas imagens são extraídos os atributos utilizados para formar o vetor de características e servir como entrada da rede neural para a classificação. Os mesmos atributos utilizados para o treinamento são os utilizados para a classificação. Quando

Tabela III
RESULTADOS DA CLASSIFICAÇÃO CONSIDERANDO DIFERENTES COMBINAÇÕES DE PARÂMETROS/ATRIBUTOS. FONTE: DADOS DA PESQUISA.

densidade	*altura*	*largura*	*soma linhas*	*soma colunas*	*algoritmo*	% acerto
1	0	0	0	0		87,3%
1	1	0	0	0		88,7%
1	0	1	0	0		90,3%
1	1	1	0	0		**90,1%**
1	0	0	0	1	binario	87,6%
1	0	0	0	1	z-pixel	88,6%
1	0	0	1	0	binario	84,8%
1	0	0	1	0	z-pixel	83,6%
1	0	0	1	1	binario	84,7%
1	0	0	1	1	z-pixel	84,6%
1	1	1	1	1	binario	87,3%
1	1	1	1	1	z-pixel	77,8%

mudam-se os atributos para treinamento, automaticamente mudamos os atributos que serão utilizados na classificação.

No final da classificação, um arquivo é salvo com o resultado da classificação (número de acertos, número de erros, percentual de acerto) e a matriz de confusão.

D. Testes realizados

Foram escolhidos vários parâmetros para variar, porém se fossem combinados todos, seria necessário executar milhares de testes diferentes, o que tornaria o processo inviável. Portanto, foram diminuídos o número de combinações, considerando os melhores resultados de cada variação de parâmetro. Os resultados são apresentados na Tabela III. Nesta tabela são apresentadas as variações na utilização ou não de atributos (0: não utilizado; 1: utilizado) e a escolha dos algoritmos *binario* ou *z-pixel* para a contagem de pixels dos objetos.

Na Tabela IV é apresentada a matriz de confusão do melhor resultado obtido, $90,3\%$. Este resultado foi obtido utilizando os atributos densidade e largura.

Tabela IV
MATRIZ DE CONFUSÃO DO MELHOR RESULTADO OBTIDO. FONTE: DADOS DA PESQUISA.

	apple	banana	cell phone	flashlight	lightbulb	soda can
apple	**1164**	0	6	0	92	0
banana	0	**661**	0	17	6	3
cell phone	2	48	**780**	17	193	23
flashlight	0	67	3	**1066**	0	11
lightbulb	1	0	29	1	**589**	14
soda can	0	20	4	0	0	**1150**

IV. CONCLUSÃO

Como não foi usada a cor dos objetos (RGB), sendo usada apenas a profundidade (*Depth*), os resultados de classificação poderiam ainda ser melhorados pela inclusão desta informação (aumento da quantidade de informações disponíveis sobre os objetos). O acerto de $90,3\%$ baseado apenas nas informações de profundidade pode ser considerado um resultado relevante em relação aos demais métodos atualmente existentes, inclusive que utilizam a mesma base de dados UW RGB-D, conforme apresentados na Tabela V. Porém, uma vez que não foi usada a cor, este método pode ser utilizado para o reconhecimento de objetos inclusive no escuro, onde é possível obter a profundidade dos objetos com o Kinect, porém não sendo possível obter atributos de cor (RGB).

Tabela V
COMPARAÇÃO DOS RESULTADOS RECENTES DA BASE DE DADOS DA UW. FONTE: ADAPTADA DE [5].

Métodos supervisionados	Autor	*Depth*	RGB	Ambos
linear svm	[14]	53,1	74,3	81,9
kernel svm	[14]	64,7	74,5	83,8
random forest	[14]	66,8	74,7	79,6
IDL	[17]	70,2	78,6	85,4
3D SPMK	[18]	67,8	-	-
KDES	[19]	78,8	77,7	86,2
CKM	[20]	-	-	86,4
HMP	[21]	70,3	74,7	82,1
SP-HMP	[22]	81,2	82,4	87,5
CNNRNN	[23]	78,9	80,8	86,8
CNN	[24]	-	83,1	89,4
R^2ICA	[25]	83,9	85,7	89,6
FusCNN(HHA)	[26]	83	84,1	91
FusCNN(jet)	[26]	83,8	84,1	91,3
warping	[27]	-	-	92,7
NMSS	[28]	75,6	74,6	88,5
CFK	[29]	85,8	86,8	91,2
Métodos semi-supervisionados	Autor	*Depth*	RGB	Ambos
CT+SVM$_1$	[30]	71,8	77,1	81,6
CT+SVM$_2$	[31]	75,4	78,7	83,7
DCNN	[5]	82,6	85,5	89,2

V. TRABALHOS FUTUROS

As seguintes melhorias e trabalhos futuros que estão sendo desenvolvidos junto a este projeto, incluem:

1) Desenvolver outros algoritmos para extração de características das imagens;
2) Realizar mais testes com mudanças de parâmetros de extração de atributos e parâmetros diferentes da rede neural;
3) Aplicar esta classificação utilizando as 51 classes.

AGRADECIMENTOS

Os autores agradecem ao ICMC/USP, LRM/ICMC/USP, INCT-SEC pela infraestrutura disponibilizada para realização deste trabalho, bem como à CAPES e ao CNPq pelo apoio financeiro durante este trabalho.

REFERÊNCIAS

[1] E. Prado, J. L. A. Samatelo, P. M. Ciarelli, e W. H. Hisatugu, "Reconhecimento de imagens: Um novo aliado do diagnóstico Digital na Medicina", Convergência Digital, Março, 2016. Disponível em: http://www.convergenciadigital.com.br/cgi/cgilua.exe/sys/start.htm?UserActiveTemplate=site&infoid=41782&sid=15. Acesso em 30 de abril de 2018.

[2] R. Romero, E. Prestes Jr, F. S. Osório, e D. F. Wolf, Robótica Móvel, Editora LTC – Grupo GEN, Rio de Janeiro, 2013.

[3] A. H. M. Pinto, "Um sistema de reconhecimento de objetos incorporado a um robô humanoide com aplicação na educação", Dissertação de mestrado (Ciências de Computação e Matemática Computacional), ICMC/USP, São Carlos, 2015.

[4] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd. ed., Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[5] Y. Cheng, X. Zhao, R. Cai, Z. Li, K. Huang, e Y. Rui, "Semi-supervised multimodal deep learning for RGB-D object recognition", In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16), Gerhard Brewka (Ed.). AAAI Press 3345-3351, 2016.

[6] C. A. B. Przewodowski Filho, "Feature extraction from 3D point clouds", Dissertação de mestrado (Ciências de Computação e Matemática Computacional), ICMC/USP, São Carlos, 2018.

[7] C. A. B. Przewodowski Filho, e F. S. Osório, "Complex network shape descriptor for 3D objects classification", 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Curitiba, 2017, pp. 1-5.

[8] C. A. B. Przewodowski Filho, e F. S. Osório, "Co-occurrence matrices for 3D shape classification", 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Curitiba, 2017, pp. 1-5.

[9] D. O. Sales, "Extração de features 3D para o reconhecimento de objetos em nuvem de pontos", Tese de doutorado (Ciências de Computação e Matemática Computacional), ICMC/USP, São Carlos, 2017.

[10] D. O. Sales, J. Amaro, e F. S. Osório, "3D shape descriptor for objects recognition", 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Curitiba, 2017, pp. 1-6.

[11] L. Sun, C. Zhao, e R. Stolkin, "Weakly-supervised DCNN for RGB-D Object Recognition in Real-World Applications Which Lack Large-scale Annotated Training Data", arXiv, 2017. Disponível em: https://arxiv.org/abs/1703.06370 Acesso em: 30 de abril de 2018.

[12] K. Lai, L. Bo, X. Ren, e D. Fox, "A scalable tree-based approach for joint object and pose recognition", In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI'11), AAAI Press 1474-1480, 2011.

[13] C. K. Miyazaki, "Redes Neurais Convolucionais para aprendizagem e reconhecimento de objetos 3D", Dissertação de mestrado (Ciências de Computação e Matemática Computacional), ICMC/USP, São Carlos, 2017.

[14] K. Lai, L. Bo, X. Ren, e D. Fox, "A Large-Scale Hierarchical Multi-View RGB-D Object Dataset", In IEEE International Conference on Robotics and Automation (ICRA), Maio, 2011.

[15] OpenCV - Open Source Computer Vision Library. Disponível em: https://opencv.org/ Acesso em: 01 de maio de 2018.

[16] FANN – Fast Artificial Neural Network Library. Disponível em: http://leenissen.dk/fann/wp/ Acesso em: 01 de maio de 2018.

[17] K. Lai, L. Bo, X. Ren, e D. Fox "Sparse distance learning for object recognition combining rgb and depth information", In ICRA, 2011.

[18] C. R. Cabrera, R. J. L. Sastre, J. A. Rodriguez, S. M. Bascón, "Surfing the point clouds: selective 3d spatial pyramids for category-level object recognition", In CVPR, 2012.

[19] L. Bo, X. Ren, e Dieter Fox, "Depth kernel descriptors for object recognition", In IROS, 2011.

[20] M. Blum, J. T. Springenberg, J. Wulfing, e M. Riedmiller, "A learned feature descriptor for object recognition in rgb-d data", In ICRA, 2012.

[21] L. Bo, X. Ren, e D. Fox, "Hierarchical matching pursuit for image classification: architecture and fast algorithms", In NIPS, 2011.

[22] L. Bo, X. Ren, e D. Fox, "Unsupervised feature learning for rgb-d based object recognition", ISER, June, 2012.

[23] R. Socher, B. Huval, B. Bath, C. D Manning, e Andrew Ng, "Convolutional-recursive deep learning for 3d object classification", In NIPS, 2012.

[24] M. Schwarz, H. Schulz, e S. Behnke, "Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features", In ICRA, 2015.

[25] I-H. Jhuo, S. Gao, L. Zhuang, DT Lee, e Y. Ma, "Unsupervised feature learning for rgb-d image classification", In ACCV, 2015.

[26] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, e W. Burgard, "Multimodal deep learning for robust rgb-d object recognition", IROS, 2015.

[27] Y. Cheng, R. Cai, C. Zhang, Z. Li, X. Zhao, K. Huang, e Y. Rui, "Query adaptive similarity measure for rgb-d object recognition", In ICCV, 2015.

[28] A. Wang, J. Cai, J. Lu, e T.-J. Cham, "Mmss: Multi-modal sharable and specific feature learning for rgb-d object recognition", In ICCV, 2015.

[29] Y. Cheng, R. Cai, X. Zhao, e K. Huang. "Convolutional fisher kernels for rgb-d object recognition", In 3DV, 2015.

[30] Y. Cheng, X. Zhao, K. Huang, e T. Tan, "Semi-supervised learning for rgb-d object recognition", In ICPR, 2014.

[31] Y. Cheng, X. Zhao, K. Huang, e T. Tan, "Semi-supervised learning and feature evaluation for rgb-d object recognition", Computer Vision and Image Understanding, 139:149-160, 2015.