IEEE WCCI 2020

IEEE World Congress on Computational Intelligence

Virtual Conference – July 19-24, 2020

VISUALLY IMPAIRED AID USING CONVOLUTIONAL NEURAL NETWORKS, TRANSFER LEARNING, AND PARTICLE COMPETITION AND COOPERATION

Fabricio A. Breve and Carlos N. Fischer

São Paulo State University (UNESP)

Motivation

- It is estimated that at least 2.2 billion people have a vision impairment or blindness. [1]
- The majority of them are over 50 years old and live in low and middle-income regions. [2]
- Navigation and mobility are among the most critical problems faced by visually impaired persons.

[1] World Health Organization, “Vision impairment and blindness,” Oct 2019, accessed: 2019-01-14. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[2] R. R. Bourne, S. R. Flaxman, T. Braithwaite, M. V. Cicinelli, A. Das, J. B. Jonas, J. Keeffe, J. H. Kempen, J. Leasher, H. Limburg et al., “Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis,” The Lancet Global Health, vol. 5, no. 9, pp. e888–
e897, 2017.

[3] C. K. Lakde and P. S. Prasad, “Review paper on navigation system for visually impaired people,” International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, no. 1, 2015.

- There were many advances in computer vision and some proposed navigation systems in the last decade.
- **Issues**: Many of them *require expensive, heavy, and/or not broadly available equipment, or require a network connection to a powerful remote server*.
- The white cane is still the most popular, simplest tool for detecting obstacles due to its low cost and portability. [3]

Objectives

- **Project Goal**: build a system to assist visually impaired people.
- **Requirement**: execute on a single smartphone, without extra accessories or connection requirements.
- **Method**: the smartphone takes pictures of the path and provides audio and/or vibration feedback regarding potential obstacles, before they are in the reach of the white cane.
- **This Paper Goal:** build the classification step, based on:
  - Convolutional Neural Networks (CNNs);
  - Transfer Learning (TL);
  - Semi-Supervised Learning (SSL) using the Particle Competition and Cooperation (PCC) method.

Why Convolutional Neural Networks?

- CNNs training phase commonly has very high computational costs.
- However, once trained, CNNs are relatively fast to make inferences.
- Most current smartphones SoCs (System-on-a-Chip) are able to make inferences on a single image using CNN models, like VGG19, in the range of milliseconds. [24]

[24] A. Ignatov and R. Timofte, "Ai benchmark: All about deep learning on smartphones in 2019," in IEEE International Conference on Computer Vision (ICCV) Workshops, 2019

The Dataset

- We propose a dataset with:
  - 342 images;
  - Two classes:
    - 175 images of “clear-path”;
    - 167 images of “non clear-path”;

- The Dataset covers:
  - Indoor and outdoor situations;
  - Different types of floor;
  - Dry and wet floor;
  - Different amounts of light;
  - Daylight and artificial light;
  - Different types of obstacles:
    - Stairs, trees, holes, animals, traffic cones, etc.

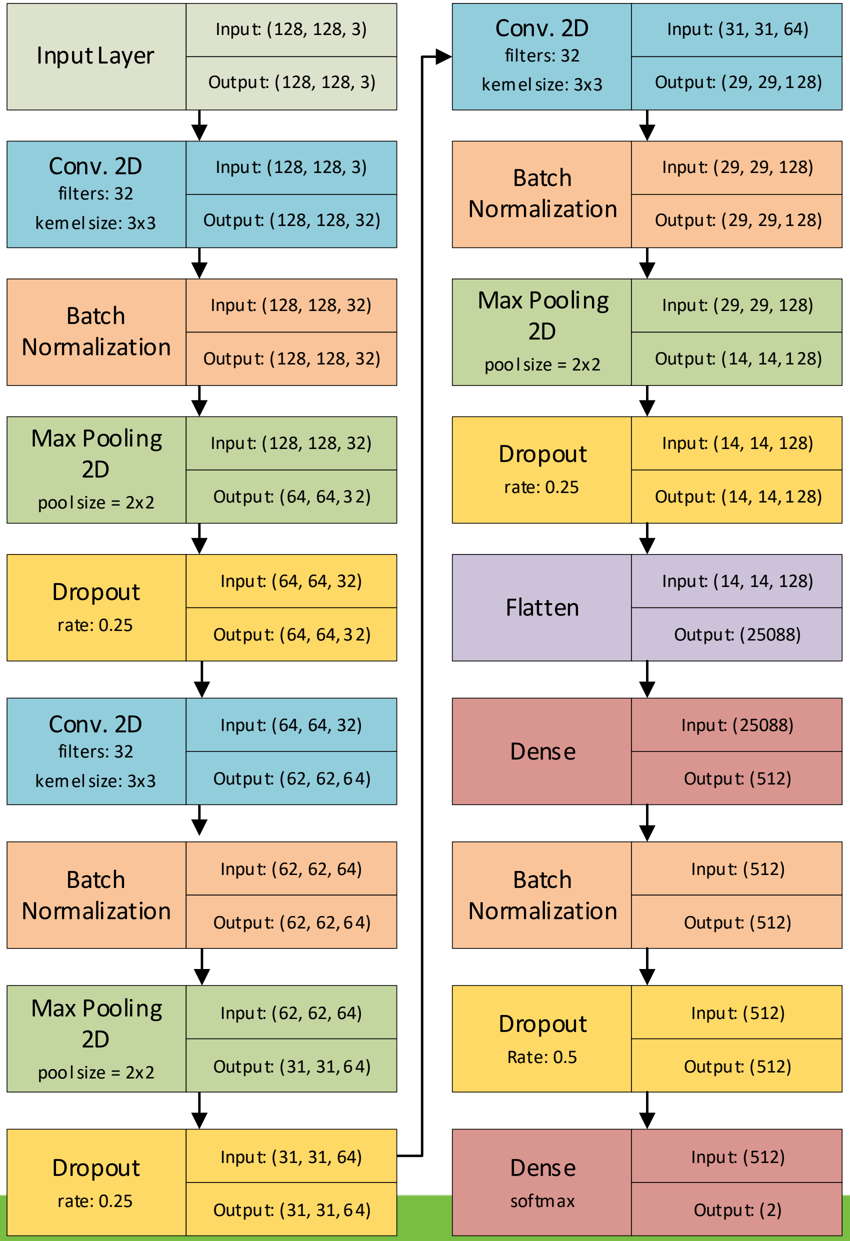
Clear Path

Non-Clear Path

The full dataset is available at: https://github.com/fbreve/via-dataset

Baseline CNN

- Input images resized to 128 x 128 pixels
- 3 Convolutional Layers.
  - Each followed by normalization and max-polling layers.
- Dense Intermediate Layer.
  - Followed by normalization and dropout layers.
- Classification Layer.
- ReLU activation functions
  - Except classification layer (softmax)
- Data Augmentation
- Different optimizers
  - Adam, RMSProp, SGD



Baseline CNN

- Results:

Data Augmentation	Optimizer	Accuracy	
No	Adam	67.97%	± 7.33%
No	RMSProp	70.35%	± 8.67%
No	SGD	60.53%	± 8.64%
Yes	Adam	73.51%	± 7.98%
Yes	RMSProp	**76.40%**	**± 7.14%**
Yes	SGD	72.19%	± 7.57%

Transfer Learning

- 17 architectures trained pre-trained in the ImageNet dataset [25]
- Four different scenarios:
  - Frozen Layers vs. Tunable Layers
  - No Polling vs. Global Average Polling

VGG16	Input: (224, 224, 3)	Output: (7,7,512)
Flatten	Input: (7, 7,512)	Output: (25088)
Dense	Input: (25088)	Output: (128)
Dense softmax	Input: (128)	Output: (2)

Example: VGG16 and No Polling

VGG16	Input: (224, 224, 3)	Output: (7,7,512)
Global Pooling 2D	Input: (7, 7,512)	Output: (512)
Dense	Input: (512)	Output: (128)
Dense softmax	Input: (128)	Output: (2)

Example: VGG16 and Average Polling

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

Architecture	Frozen Weights				Fine-Tunable Weights			
	No Polling		Average Polling		No Polling		Average Polling	
Xception	49.43%	± 6.82	51.18%	± 8.61	**87.70%**	**± 2.62%**	**92.11%**	**± 5.01%**
VGG16	**83.39%**	**± 13.35%**	**76.88%**	**± 7.32%**	85.76%	± 11.97%	85.16%	± 12.96%
VGG19	81.36%	± 11.11%	73.97%	± 5.53%	83.40%	± 11.38%	85.18%	± 13.29%
ResNet50	51.17%	± 8.82%	51.18%	± 8.61%	49.98%	± 9.31%	51.77%	± 8.77%
ResNet101	51.18%	± 8.61%	50.87%	± 8.08%	66.12%	± 19.75%	51.42%	± 8.83%
ResNet152	48.05%	± 10.97%	51.17%	± 7.88%	54.90%	± 12.51%	47.37%	± 4.85%
ResNet50V2	51.19%	± 11.08%	51.19%	± 11.08%	51.48%	± 8.72%	69.71%	± 22.50%
ResNet101V2	51.18%	± 8.61%	51.18%	± 8.12%	63.49%	± 9.02%	51.46%	± 9.02%
ResNet152V2	51.18%	± 8.12%	53.25%	± 11.63%	54.40%	± 7.53%	54.69%	± 6.24%
InceptionV3	51.18%	± 8.61%	51.18%	± 8.61%	79.94%	± 16.02%	88.90%	± 3.38%
InceptionResNetV2	51.18%	± 8.12%	51.19%	± 11.08%	75.53%	± 11.90%	78.37%	± 5.08%
MobileNet	51.48%	± 8.72%	51.18%	± 8.61%	81.43%	± 16.66%	90.08%	± 5.02%
DenseNet121	51.18%	± 8.61%	51.18%	± 8.61%	54.71%	± 9.75%	45.03%	± 8.62%
DenseNet169	51.45%	± 8.04%	48.24%	± 8.73%	64.63%	± 12.02%	75.50%	± 15.18%
DenseNet201	48.34%	± 10.90%	51.46%	± 9.11%	61.80%	± 17.82%	59.34%	± 8.70%
NASNetMobile	51.78%	± 9.43%	49.13%	± 12.40%	50.29%	± 8.46%	49.13%	± 8.80%
MobileNetV2	50.90%	± 8.65%	51.19%	± 11.08%	50.90%	± 8.65%	51.18%	± 8.61%

VGG16: different amount of frozen layers

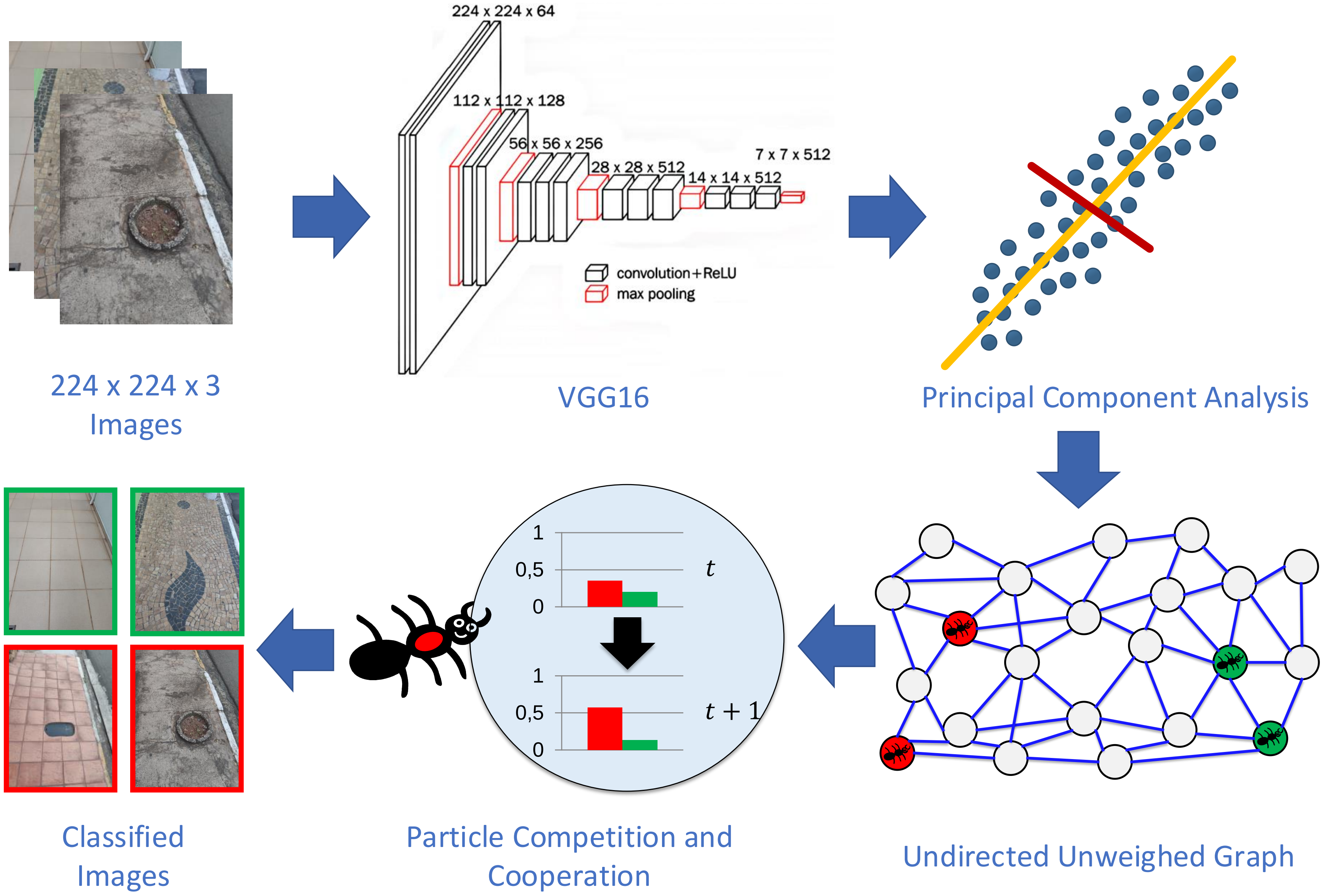
Frozen Layer Blocks	No Polling		Average Polling	
None	89.19%	± 6.46%	86.93%	± 6.42%
1	88.56%	± 5.84%	88.95%	± 6.84%
2	88.96%	± 5.99%	**89.23%**	**± 7.52%**
3	**89.40%**	**± 6.50%**	88.42%	± 6.84%
4	87.10%	± 5.84%	87.65%	± 7.39%
All	86.36%	± 7.09%	74.78%	± 7.00%

Xception and MobileNet: different amount of frozen layers

Frozen Layer Blocks	Xception		MobileNet	
None	**91.68%**	**± 3.58%**	**88.89%**	**± 3.36%**
1	48.26%	± 8.27%	51.17%	± 7.48%
2	49.55%	± 8.82%	52.03%	± 9.70%
3	48.95%	± 7.91%	50.74%	± 10.39%
4	48.80%	± 7.35%	49.85%	± 10.05%
5	51.18%	± 11.15%	45.13%	± 10.43%

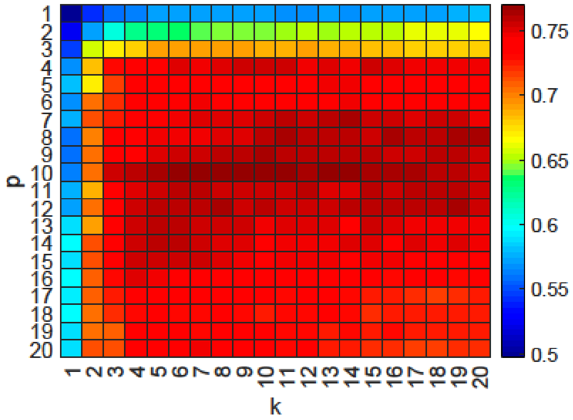
Why Semi-Supervised Learning and Particle Competition and Cooperation?

- **Problem:** It is difficult to acquire and label pictures of the many different scenarios an user may face.
- **Solution:** Incorporate new knowledge, acquired from user feedback.
- **Problem**: CNNs inference in smartphones is feasible, but the training process is not.
- **Solution:** use VGG16 and VGG19 as feature extractors and a fast SSL method, like PCC, to incorporate knowledge on-the-fly.
    - The CNN weights are frozen.
    - A graph is built from the CNN output and fed to PCC.
    - Principal Component Analysis is used to reduce dimensionality.
    - PCC can incorporate new data and new labels at low cost.

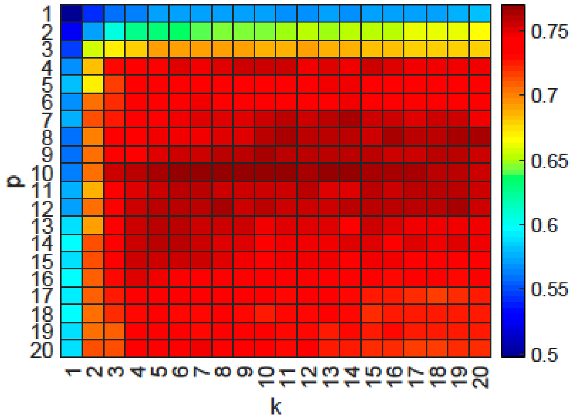

224 x 224 x 3
Images
224 x 224 x 64
112 x 112 x 128
56 x 56 x 256
28 x 28 x 512
14 x 14 x 512
7 x 7 x 512
convolution+ReLU
max pooling
VGG16
Principal Component Analysis
Undirected Unweighed Graph
1
0,5
0
t
1
0,5
0
$t+1$
Particle Competition and
Cooperation
Classified
Images

PCC Framework: best parameters

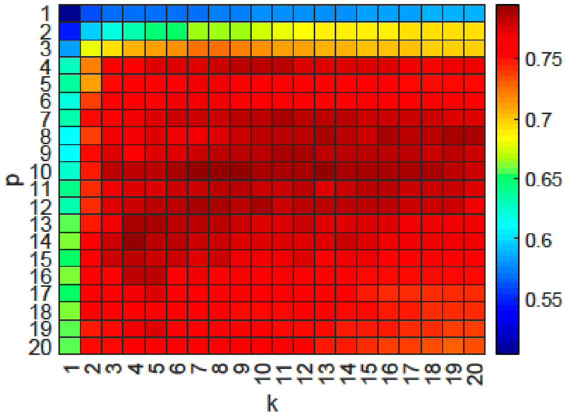
- PCC framework accuracy varying the number of:
  - *k*-nearest neighbors (graph construction)
  - *p* principal components (PCA)



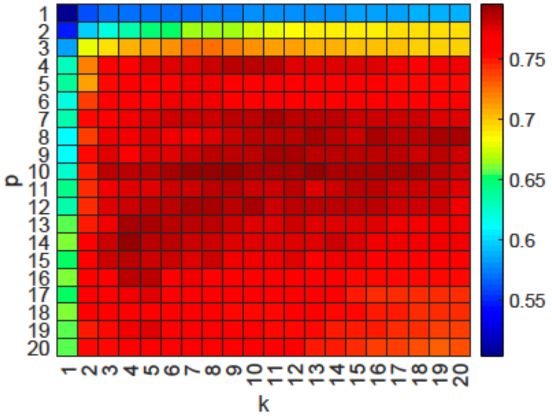
VGG19 – 10%
Labeled Nodes



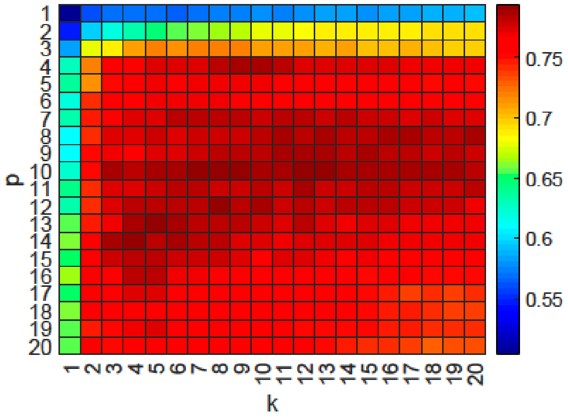
VGG19 – 20%
Labeled Nodes



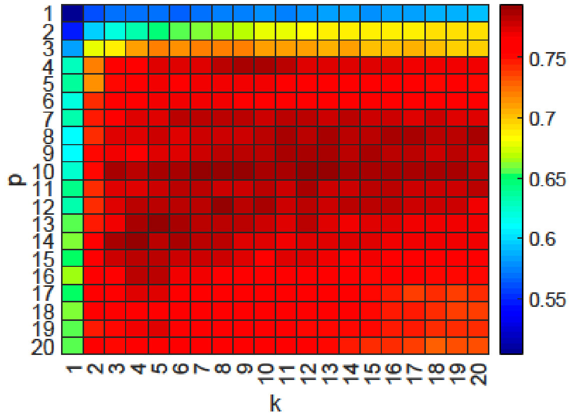
VGG16 – 10%
Labeled Nodes



VGG16 – 20%
Labeled Nodes



VGG16 + VGG19 – 10%
Labeled Nodes



VGG16 + VGG19 – 20%
Labeled Nodes

PCC Framework: Results

Labeled Nodes	Architecture	*p*	*k*	Accuracy	
10%	VGG16	10	7	**77.01%**	**± 3.55%**
10%	VGG19	10	8	76.99%	± 3.60%
10%	VGG16+VGG19	10	8	76.99%	± 3.68%
20%	VGG16	10	7	**79.53%**	**± 2.40%**
20%	VGG19	10	8	79.35%	± 2.65%
20%	VGG16+VGG19	14	4	79.43%	± 2.65%

Conclusions

- We propose methods to help in identifying obstacles in the path of visually impaired people.
  - These methods have low computational costs in the inference step.
    - Milliseconds in current smartphones;
    - They can be implemented without relying on other equipment or remote servers.
- We also propose a dataset to help in the training of these methods.
- We compared many consolidated CNN architectures pre-trained on a large dataset and fine-tuned them to the proposed task.
- We use pre-trained CNN architectures as feature extractors for semi-supervised learning classification.
  - Particle competition and cooperation method.

Conclusions

- Computer simulations showed promising results with some of the CNN architectures.
- The SSL also achieved relatively high accuracy.
  - Considering that it is using only up to 20% of the dataset for training and no fine-tuning in CNN networks.
- Future Work:
  - Acquire more images to the proposed dataset;
  - Search for other approaches and tweaks in the current framework to further improve the classification accuracy;
  - Build a smartphone prototype application to test some real-world scenarios.