Convolutional Neural Networks and Ensembles for Visually Impaired Aid

unesp

Fabricio Breve
São Paulo State University - UNESP
fabricio.breve@unesp.br

CAPES
FAPESP

Motivation

- Approximately 2.2 billion people suffer from some form of visual impairment.
  - Including at least 1 billion with moderate or severe distance vision impairment [40].
- The prevalence of distance vision impairment is significantly higher in low- and middle-income areas compared to high-income regions [34].
- This population faces numerous difficulties in their daily routines, mostly linked to **mobility and navigation**.

- With advancements in computer vision and related technologies, numerous navigation systems have been proposed.
- **Issues:** many of them:
  - require costly, bulky, and/or custom equipment;
  - are too computationally intensive to run on portable devices;
  - require a network connection to a more powerful remote server.
- White canes and guide dogs are currently the most commonly utilized tools to aid visually impaired (VI) individuals [15].

[34] Steinmetz, J.D., Bourne, R.R., Briant, P.S., Flaxman, S.R., Taylor, H.R., Jonas, J.B., Abdoli, A.A., Abrha, W.A., Abualhasan, A., Abu-Gharbieh, E.G., et al.: Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to vision 2020: the right to sight: na analysis for the global burden of disease study. The Lancet Global Health 9(2), e144-e160 (2021).

[40] World Health Organization: Vision impairment and blindness (Oct 2022), https://www.who.int/news-room/fact-sheets/detail/blindness-and-visualimpairment, accessed: 2023-01-30.

[15] Islam, M.M., Sheikh Sadi, M., Zamli, K.Z., Ahmed, M.M.: Developing walking assistants for visually impaired people: A review. IEEE Sensors Journal 19(8), 2814-2828 (2019). https://doi.org/10.1109/JSEN.2018.2890423.

Motivation

- Recent surveys show that:
  - Smartphone-based computer vision tools for the VI often employ **outdated** image and video processing techniques [3];
  - Researchers have started to adopt deep learning approaches [24]:
    - These techniques have grown with the advent of increased computational power in machines;
    - However, carrying high-powered computational devices for vision-based assistive solutions is **not practical** for users.

[3] Budrionis, A., Plikynas, D., Daniu²is, P., Indrulionis, A.: Smartphonebased computer vision travelling aids for blind and visually impaired individuals: A systematic review. Assistive Technology 34(2), 178194 (2022). https://doi.org/10.1080/10400435.2020.1743381, pMID: 32207640.

[24] Mandia, S., Kumar, A., Verma, K., Deegwal, J.K.: Vision-based assistive systems for visually impaired people: A review. In: Tiwari, M., Ismail, Y., Verma, K., Garg, A.K. (eds.) Optical and Wireless Technologies. pp. 163172. Springer Nature Singapore, Singapore (2023).

Objectives

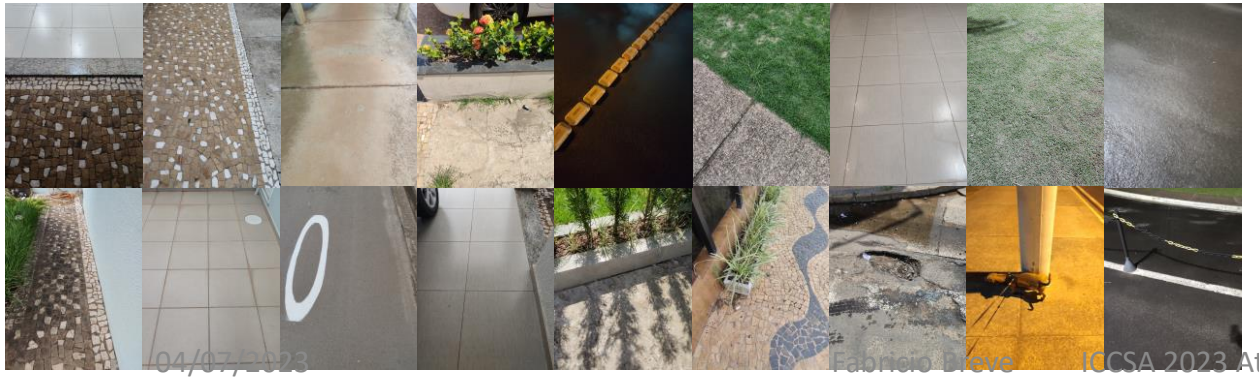
- **Project Goal**: build a system to assist visually impaired people.
- **Requirement**: execute on a single smartphone, without extra accessories or connection requirements.
- **Method**: the smartphone takes pictures of the path and provides audio and/or vibration feedback regarding potential obstacles, before they are in the reach of the white cane.
- **This Paper Goal**: perform the classification step, based on Convolutional Neural Networks (CNNs).
  - Find the best CNN architecture for this task;
  - Find the optimal learning hyperparameters.

Contributions

- **Prior study [2]**: a framework that leverages CNNs, transfer learning, and semi-supervised learning (SSL).
  - The focus was to minimize computational costs and make it feasible for implementation on smartphones without requiring additional hardware.
- **This study**: previous works are significantly expanded upon with the following key contributions:
  1. Eight additional CNN models were added to the study, based on the cutting-edge EfficientNet architecture [37], bringing the total number of networks evaluated to 25;
  2. The K-Fold Cross Validation process was repeated five times, providing more robust results;
  3. Image pre-processing functions were introduced to enhance image preparation for each network type, resulting in improved accuracy in most cases;
  4. Ensembles of CNNs were employed to boost the overall accuracy by leveraging the strengths of multiple CNN architectures.

The Dataset

- 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

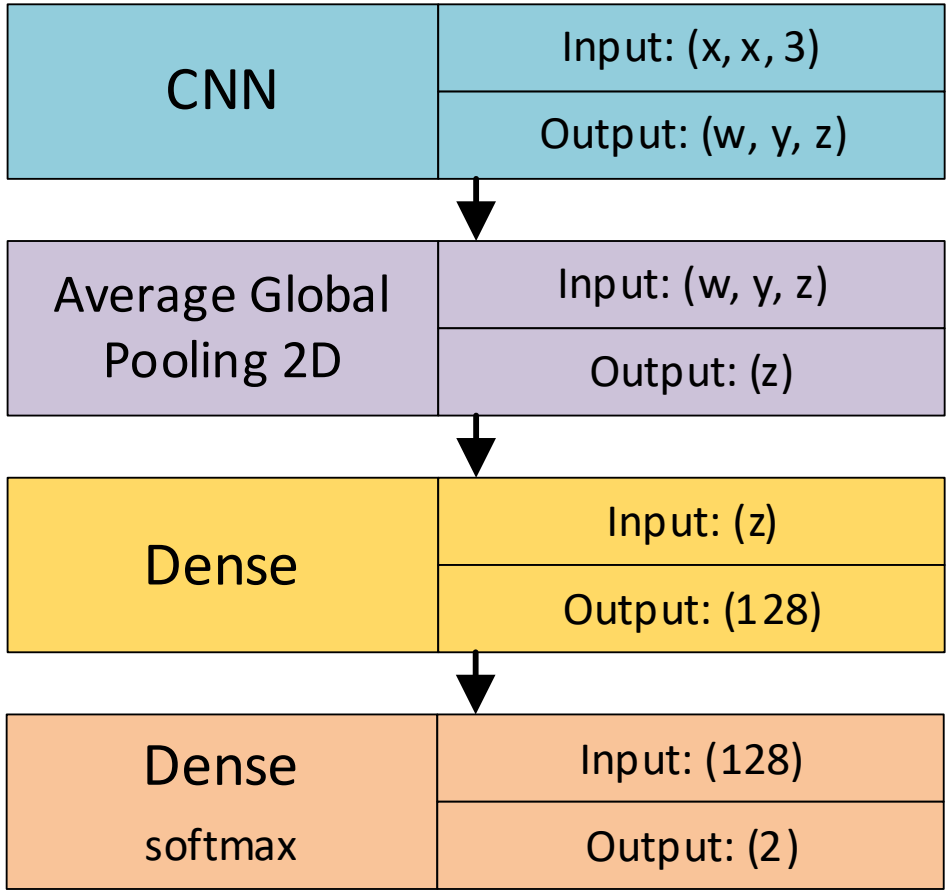
The full dataset is available at: https://github.com/fbreve/via-dataset

CNN Architectures

- 25 evaluated architectures

Model	Input Image Resolution	Output of Last Conv. Layer	Trainable Parameters
DenseNet121	224 × 224	7 × 7 × 1024	7,085,314
DenseNet169	224 × 224	7 × 7 × 1664	12,697,858
DenseNet201	224 × 224	7 × 7 × 1920	18,339,074
EfficientNetB0	224 × 224	7 × 7 × 1280	4,171,774
EfficientNetB1	240 × 240	8 × 8 × 1280	6,677,410
EfficientNetB2	260 × 260	9 × 9 × 1408	7,881,604
EfficientNetB3	300 × 300	10 × 10 × 1536	10,893,226
EfficientNetB4	380 × 380	12 × 12 × 1792	17,778,378
EfficientNetB5	456 × 456	15 × 15 × 2048	28,603,314
EfficientNetB6	528 × 528	17 × 17 × 2304	41,031,002
EfficientNetB7	600 × 600	19 × 19 × 2560	64,115,026
InceptionResNetV2	299 × 299	8 × 8 × 1536	54,473,186
InceptionV3	299 × 299	8 × 8 × 2048	22,030,882
MobileNet	224 × 224	7 × 7 × 1024	3,338,434
MobileNetV2	224 × 224	7 × 7 × 1280	2,388,098
NASNetMobile	224 × 224	7 × 7 × 1056	4,368,532
ResNet101	224 × 224	7 × 7 × 2048	42,815,362
ResNet101V2	224 × 224	7 × 7 × 2048	42,791,426
ResNet152	224 × 224	7 × 7 × 2048	58,482,050
ResNet152V2	224 × 224	7 × 7 × 2048	58,450,434
ResNet50	224 × 224	7 × 7 × 2048	23,797,122
ResNet50V2	224 × 224	7 × 7 × 2048	23,781,890
VGG16	224 × 224	7 × 7 × 512	14,780,610
VGG19	224 × 224	7 × 7 × 512	20,090,306
Xception	299 × 299	10 × 10 × 2048	21,069,482

Proposed CNN Networks



- Weights initialization:
  - **Convolutional layers**: pre-trained weights from the Imagenet dataset [28].
  - **Dense layers**: He uniform variance scaling initializer [7].

[28] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115(3), 211252 (2015). https://doi.org/10.1007/s11263-015-0816-y.

[7] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classication. In: Proceedings of the IEEE international conference on computer vision. pp. 10261034 (2015).

Tested Scenarios

Config.	Fine-Tuning	Different Learning Rates	Optimizer
A	No	No	RMSprop
B	No	No	Adam
C	Yes	No	RMSprop
D	Yes	No	Adam
E	Yes	Yes	RMSprop
F	Yes	Yes	Adam

- Scenarios A to D:
  - Adaptive learning rate: 10^{-3} to 10^{-5}.
    - Adjustment factor: 0.5 when validation accuracy did not increase in the last two epochs.
- Scenarios E and F:
  - Fixed learning rate:
    - Dense layers: 10^{-3}; Convolutional layers: 10^{-5}.
- All scenarios:
  - Up to 50 epochs;
    - Early stop: validation loss did not decrease in the last 10 epochs.

Tested Scenarios

- **Software**: Python with TensorFlow.
- **Hardware**: 3 desktop computers equipped with NVIDIA GeForce GPU boards:
  - GTX 970;
  - GTX 1080;
  - RTX 2060 SUPER.
- The code is available at GitHub:
  - https://github.com/fbreve/via-py

- K-Fold Cross Validation:
  - $k = 10$
  - 5 repetitions
- Validation subset:
  - 20% of the training instances.
- Batch size: 16
  - Exceptions:

Method	Conf. C	Conf. D	Conf. E	Conf. F
EfficientNetB3			16 - 8	
EfficientNetB4	4	8	4	8
EfficientNetB5	2	4	2	4
EfficientNetB6	2	2	2	2
EfficientNetB7	1	1	1	1

Results: Single Networks

Method	Conf. A	Conf. B	Conf. C	Conf. D	Conf. E	Conf. F	Average
MobileNet	**0,9158**	**0,9152**	*0,9299*	0,9257	0,8729	**0,9105**	**0,9117**
Xception	0,8749	0,8755	*0,9374*	0,9252	0,8934	0,9029	0,9016
EfficientNetB0	0,8877	0,8876	***0,9427***	0,9274	0,8724	0,8819	0,9000
EfficientNetB3	0,8901	0,8889	*0,9404*	0,9291	0,8727	0,8761	0,8995
EfficientNetB2	0,8725	0,8660	0,9391	*0,9426*	0,8672	0,8679	0,8926
EfficientNetB4	0,8813	0,8807	0,9304	***0,9456***	0,8414	0,8632	0,8904
EfficientNetB1	0,8908	0,8855	*0,9369*	0,9341	0,8482	0,8463	0,8903
DenseNet201	0,8841	*0,8847*	*0,8847*	0,8807	0,8696	0,8809	0,8808
InceptionResNetV2	0,8650	0,8644	0,8954	*0,9217*	0,8632	0,8668	0,8794
InceptionV3	0,8691	0,8657	0,8611	*0,8965*	**0,8936**	0,8807	0,8778
DenseNet169	0,8789	0,8766	0,8779	*0,8854*	0,8679	0,8713	0,8763
DenseNet121	0,8730	0,8671	0,8867	*0,8912*	0,8715	0,8680	0,8763
ResNet50	*0,8947*	0,8901	0,8139	0,8392	0,8801	0,8761	0,8657
ResNet101	*0,8925*	0,8919	0,8130	0,7919	0,8731	0,8626	0,8542
EfficientNetB5	0,8953	0,8947	0,8760	*0,9233*	0,6398	0,8327	0,8436
MobileNetV2	*0,8924*	0,8912	0,8324	0,7954	0,8116	0,8042	0,8378
ResNet152	0,8755	0,8754	0,7418	0,7724	*0,8819*	0,8638	0,8351
ResNet50V2	0,8539	0,8581	0,7273	0,8263	0,8516	*0,8587*	0,8293
EfficientNetB6	0,8719	0,8690	0,8514	*0,8738*	0,7383	0,7516	0,8260
ResNet101V2	*0,8807*	0,8790	0,6184	0,7256	0,8778	0,8779	0,8099
ResNet152V2	*0,9106*	0,9077	0,5942	0,6663	0,8890	0,8885	0,8094
VGG19	0,8263	0,8118	0,7916	0,6707	*0,8746*	0,8680	0,8072
VGG16	0,8263	0,8175	0,7877	0,6316	*0,8759*	0,8543	0,7989
NASNetMobile	0,8560	*0,8578*	0,6671	0,6997	0,7092	0,7050	0,7491
EfficientNetB7	*0,8731*	0,8714	0,5248	0,4999	0,5242	0,5230	0,6361
Average	*0,8773*	0,8749	0,8241	0,8289	0,8344	0,8433	0,8472

Ensembles

- Average of ensemble members for each of the 50 folds ($k = 10$, repeated 5 times).
  - Output of the last dense layer, before softmax, taken as probabilities for each class.
- The best model for each of the 6 configurations were used.
- Two ensemble experiments:
  - Multiple instances of the same model.
    - Randomized: seeds, initial weights, validation subset, etc.
  - From 1 to 10 instances of the best models in each configuration.
    - Best 2 to 6 configurations.

Results: Ensembles of Single CNN Models

Instances	Conf. A	Conf. B	Conf. C	Conf. D	Conf. E	Conf. F	Average
1	0,9158	0,9152	0,9427	*0,9456*	0,8936	0,9105	0,9206
2	0,9187	0,9135	0,9451	*0,9502*	0,9065	0,9170	0,9252
3	0,9170	0,9176	0,9445	*0,9532*	0,9112	0,9193	0,9271
4	0,9164	0,9158	0,9485	*0,9562*	0,9176	0,9182	0,9288
5	0,9187	0,9182	0,9491	*0,9585*	0,9171	**0,9217**	0,9306
6	0,9176	**0,9199**	**0,9549**	***0,9602***	0,9182	0,9199	**0,9318**
7	**0,9211**	0,9182	0,9543	*0,9579*	**0,9211**	0,9164	0,9315
8	0,9164	0,9158	0,9509	*0,9596*	0,9194	0,9158	0,9297
9	0,9182	0,9176	0,9538	*0,9596*	0,9200	0,9193	0,9314
10	0,9188	0,9159	0,9526	***0,9602***	0,9194	0,9176	0,9308
Average	0,9179	0,9168	0,9496	*0,9561*	0,9144	0,9176	0,9287

Results: Ensembles of Multiple CNN Models

Instances of each conf.	Conf. D	Conf. DC	Conf. DCA	Conf. DCAB	Conf. DCABF	All Conf.	Average
1	0,9456	0,9532	*0,9567*	0,9550	0,9503	0,9491	0,9517
2	0,9502	0,9491	*0,9544*	0,9538	0,9486	0,9521	0,9514
3	0,9532	0,9544	*0,9597*	0,9545	0,9509	0,9539	0,9544
4	0,9562	0,9555	*0,9614*	0,9527	0,9515	0,9556	0,9555
5	0,9585	0,9567	*0,9620*	0,9544	0,9544	**0,9573**	0,9572
6	**0,9602**	**0,9579**	***0,9655***	0,9562	0,9538	0,9556	**0,9582**
7	0,9579	0,9544	*0,9643*	**0,9574**	0,9544	0,9550	0,9572
8	0,9596	0,9532	*0,9626*	0,9562	0,9526	**0,9573**	0,9569
9	0,9596	0,9555	*0,9637*	0,9568	0,9568	0,9550	0,9579
10	**0,9602**	0,9561	*0,9626*	0,9556	**0,9579**	0,9550	0,9579
Average	0,9561	0,9546	*0,9613*	0,9553	0,9531	0,9546	0,9558

Conclusions

- Comparison of 25 different CNN architectures to identify obstacles in the path of visually impaired individuals.
- K-Fold Cross Validation was utilized with $k = 10$ and five repetitions to provide robust results.
- Architectures have low computational costs during inference, executing in milliseconds on current smartphones.
  - can be implemented without relying on external equipment or remote servers.
- Fine-tuning an EfficientNetB4 network achieved the highest accuracy of **0.9456**.
  - Improved to **0.9602** using an ensemble with six instances of the same network.
  - Further increased to **0.9655** by adding six instances of fine-tuned EfficientNetB0 and six instances of MobileNet with fixed weights in the convolutional layers.

Conclusions

- The numerous computer simulations conducted in this study yielded promising results for some CNN architectures and investigated the use of:
  - different optimizers (Adam and RMSprop);
  - different learning strategies (single learning rate versus different rates for convolution and dense layers);
  - fixed versus fine-tuned pre-trained weights.
- Future work:
  - expanding the proposed dataset by acquiring more images;
  - exploring other approaches and modifications to the current framework to further enhance classification accuracy.

Convolutional Neural Networks and Ensembles for Visually Impaired Aid

unesp

Fabricio Breve
São Paulo State University - UNESP
fabricio.breve@unesp.br

CAPES
FAPESP