Deep Convolutional Neural Network applied to Chagas Disease Parasitemia Assessment

André da Silva Pereira
UFRJ - IBCCF - Laboratory of Cognitive Physiology
Rio de Janeiro, RJ - Brazil
andresilper@biof.ufrj.br

Alexandre dos Santos Pyrrho, Daniel Figueiredo Vanzan
UFRJ - Faculty of Pharmacy
Laboratory of Immunoparasitology and Toxicological Analysis
Rio de Janeiro, RJ - Brazil

Leonardo Oliveira Mazza, José Gabriel R. C. Gomes
UFRJ - COPPE
Electrical Engineering Program
Rio de Janeiro, RJ - Brazil

***Abstract* — Chagas Disease is a tropical parasitic disease endemic to Latin America, and it is caused by *Trypanosoma cruzi*. It occurs in two phases. The acute phase takes place shortly after infection, and it is characterized by fever, lymphadenopathy, and chagoma symptoms. The chronic phase, which happens from a few months up to several years after infection, is generally asymptomatic, but it may also be associated with megacolon, megaesophagus, or cardiomegaly symptoms. Other heart illness symptoms may be present as well. In the acute phase, standard diagnosis is based on *T. cruzi* visualization through microscopy applied to blood smear slides. In the present work, we apply a deep convolutional neural network (namely, a pre-trained Mobile NetV2 feature extractor followed by a fine-tuned single-neuron top classifier) to the binary classification of image tiles of size 224 × 224 × 3, which are extracted from acute-phase blood smear samples. The data set corresponds to blood smear sample images taken from twelve different slides. We achieve 96.4% accuracy on a balanced validation subset within the twelve-slide data set. The respective precision, sensitivity, and F1-score values are 95.4%, 97.6%, and 96.5%. In a cross-validation experiment with five folds inside the twelve-slide data set, validation accuracy varies from 88% to 98%. From image tiles extracted from a thirteenth blood smear slide (i.e. tiles outside the train/validation sets), we estimate test accuracy equal to 72.0%, which suggests that data set size and overtraining issues must be addressed in future work.**

Keywords — Chagas disease, Trypanosoma cruzi, blood smear samples, deep convolutional neural networks.

I. Introdução

Epidemiological studies show that Chagas disease is endemic to Latin America, thus following the geographical distribution of the invertebrate hosts - insects from the Triatominae subfamily, which are popularly known as “barber bugs”. The Parasite *Trypanosoma cruzi* is Chagas disease etiological agent. The disease currently affects six million people [1], leading to approximately 14,000 annual deaths. It is considered by World Health Organization as a tropical neglected disease, because it is present among low-income populational groups, and it is subject to low research investment, medicine production, and control measures.

The onset of Chagas disease takes place in its acute phase, during which the parasites are easily pinpointed in the blood. For diagnosis based on microscopy, the blood-sample glass slides are dyed with hematological dyes (Wright or Giemsa), which renders them color varying from red to purple. Parasite size varies from 20 μm length and 1 μm width, for thin shapes, to 15 μm length and 4 μm width, for thick shapes [2]. Diagnosis during the acute phase is important, because it makes cure possible as long as treatment is started [3]. At the chronic phase, the diagnosis is based on serology, blood culture, xenodiagnosis and complementary exams such as chest x-ray or electrocardiogram [4], and treatment during the chronic phase generally does not lead to cure.

Previous works about the application of machine learning to *T. cruzi* detection are [3], [5] (Gaussian discriminant), and [6] (classifier based on k-nearest neighbors). In [3], the authors apply Adaboost and support vector machines, and report 100% sensitivity and 93.3% specificity (i.e. precision). In contrast with these conventional methods, deep convolutional networks have revolutionized computer vision in recent years, with an emphasis on object recognition [7]. Recent applications of deep learning to malaria diagnosis have been reported [8], [9]. In the present work, we propose the application of a pretrained MobileNet V2 [10] to generate features for a fine-tuned single-neuron binary classifier for automated *T. cruzi* detection in microscopy blood tests. This might help Chagas disease diagnosis mainly in non-endemic areas that have experienced disease emergence as a consequence of migratory flow [4], and might also help blood donor triage when demand becomes high, as in the case of blood banks [2].

This paper is organized as follows: in Section II, the neural network background is briefly described; the datasets and the proposed methodology are described in Section III; detailed results are presented in Section IV, and the main conclusions are presented in Section V, together with topics for future research.

II. Theoretical Foundations

The MobileNetV2 feature extractor [10] relies heavily on the concepts of depthwise separable convolutions, linear bottlenecks, and inverted residual operations, which we briefly describe in this section, for completeness. Using the depthwise separable convolutions technique, the convolution operations are split into smaller and consecutive convolutions, which reduces computational cost by a significant factor (slightly below 10) at a small accuracy penalty. In MobileNetV1 [11], the observation that features typically lie in manifolds that can be embedded in low-dimensional spaces has been used to reduce the number of dimensions in the convolutional layers. By paying close attention to the fact that the rectifying linear unit (ReLU) plays an important role in preserving information during the embedding, the authors in [10] optimize neural network structure by inserting *linear* bottleneck layers at the convolutional layers inputs. They also provide evidence against the use of non-linear activation functions for the bottleneck implementation. Finally, the linear bottleneck layers are connected by inverted residual operations which have a relatively small number of channels (i.e. a number of channels that is smaller than the number of channels of tensors that are located between the bottleneck edges). The inverted residual operations allow for computational advantages and performance similar to those reported for residual networks [7], but at lower computation cost: in [10], the authors report approximately 1/4 memory cost with respect to MobileNetV1.

With respect to the MobileNetV2 topology, its authors start with a 32-channel 3 × 3 convolutional layer at 224 × 224 × 3 input resolution, and then apply a relatively large number of convolutional layers with bottleneck operators (16 layers, some with identical configuration), eventually arriving at a 7 × 7 × 320 tensor, which is then mapped into a 7 × 7 × 1280 tensor by a 1 × 1 convolutional layer. A global average pooling operation finally maps the 7 × 7 × 1280 tensor into a 1280-component vector. The overall number of parameters reported in [10] is 3.4 million, for a basic MobileNetV2 topology, and the implementation we use in the present work, mobilenetv2_1.00_224, which is downloaded from [12], has 2.26 million parameters as reported by the model.summary() function. MobileNetV2 yielded remarkable results for image classification, object detection, and semantic segmentation. Particularly with respect to image classification, the Mobile NetV2 improved state-of-the-art performance on the ImageNet challenge [13], reporting 72.0% Top 1 accuracy with 3.4 million parameters, running in 75 ms in a cell phone (Google Pixel 1) core using TF-Lite. We use Keras [14], and the pretrained model parameters [12], in a computer with an i7 core and GPU video cards (EVGA GTX 1080 Ti). The 1280-1 classifier is trained using Adam [15], 1280-dimensional inputs computed offline, binary targets, and binary cross-entropy loss function. Section IV provides training configuration details.

III. Methodology

Blood smear slides provided by the Laboratory of Immuno-parasitology and Toxicological Analysis are observed with an optical microscope, under 1000× magnification, in immersion oil, and the microscope is connected to a camera and a computer. Images are most often (as in the case of Figure 6, without the tiling, for example) generated in TIFF format at 2592 × 1944 resolution, originally, and are then converted to JPG format for convenience. A few test images, such as the one in Figure 7, are generated at 1596 × 1198 resolution, because of camera configuration change, which is useful for classifier testing at different input image scales.

We start with twelve slides, and therefore throughout the paper we refer to the training and validation data sets as *twelve-slide data set*. From the twelve slides, we capture 1000 images at the 2592 × 1944 resolution, i.e. approximately 84 images per slide on average. Among the captured images, 208 are positive and the overall *T. cruzi* count in those images is 278. To assemble a balanced dataset for neural network training and validation, we first manually annotate, using a simple interface that was written for this annotation process, the upper-left and lower-right coordinates of all positive bounding boxes, i.e. bounding boxes containing the 278 *T. cruzi* in the images. For each positive bounding box, a negative bounding box with the same dimensions is generated at random with uniformly distributed center coordinates. Visually, we check whether the negative bounding box overlaps with any of the negative bounding boxes. If an overlap occurs, the negative bounding box is replaced by a manually generated, overlap-free, negative bounding box. This procedure leads to 556 rectangular bounding boxes (278 positive and 278 negative ones). Using an image resize Python function, the rectangular boxes are then converted into 556 square image tiles with size 224 × 224 × 3, which are compatible with the MobileNetV2 input size. Approximately 10% of the square images are removed from this set, so that they are not used for neural network training and validation, which reduces the data set to 499 square image tiles, 331 of which are used for training and 168 for validation. A binary text file containing the 499 “P” (positive) and “N” (negative) targets for each image tile is also generated. This completes the generation of the basic twelve-slide data set that is used for classifier design in Section IV. A few positive and negative image tile examples are shown in Figure 1. Variations in color, contrast, and overall aspect are clearly noticeable, although parasite sizes tend not to vary too much. A similar annotation procedure is applied to a thirteenth slide, yielding 214 additional square image tiles: 111 positive ones and 103 negative ones, as the negative bounding boxes were deleted rather than replaced in this case. A binary text file with the corresponding target is generated as well. In the present work, this additional data set is not used for classifier training or validation. Figures 9 and 10 show all 214 images.

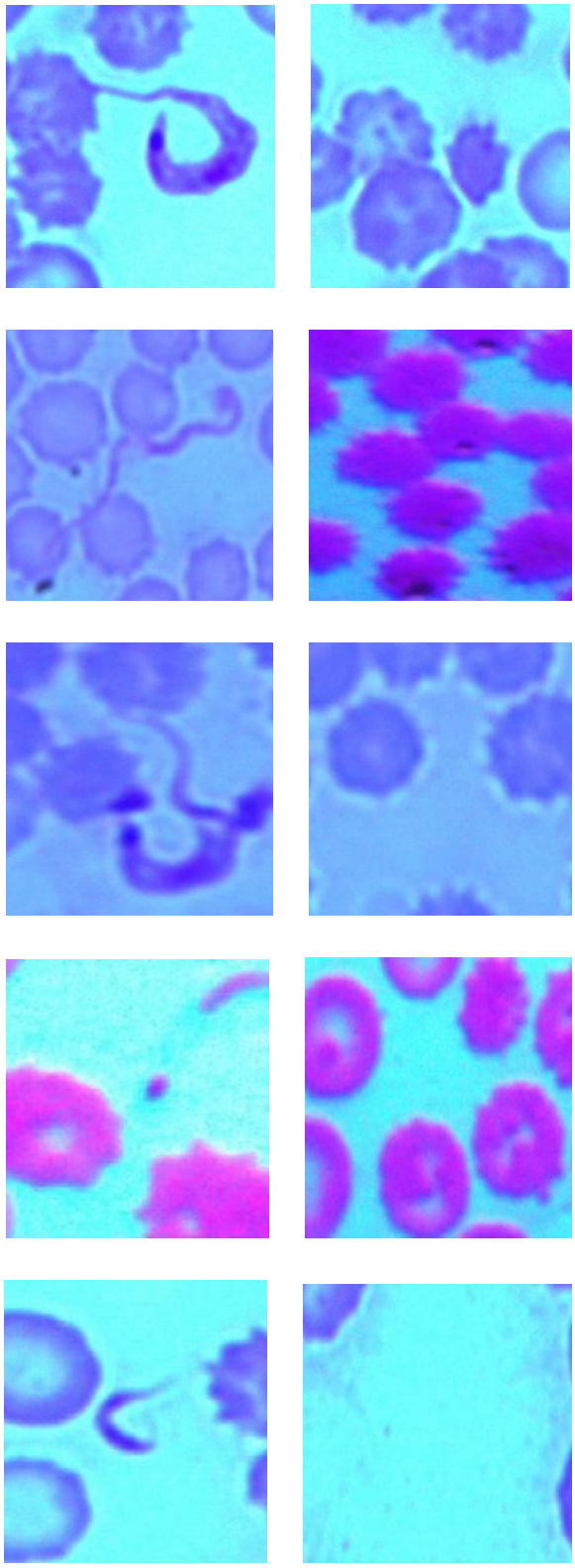


Figure 1. Positive (P, left) and negative (N, right) *T. cruzi* image tiles for neural network input. These examples are from the twelve-slide training and validation data set.

TABLE I. PRELIMINARY RESULTS WITHOUT DATA AUGMENTATION. CONFUSION MATRIX OBTAINED FROM VALIDATION DATA. TP AND TN STAND FOR TRUE-POSITIVE AND TRUE-NEGATIVE COUNTS. FP AND FN STAND FOR FALSE-POSITIVE AND FALSE-NEGATIVE COUNTS.

	Positive Target	**Negative Target**
Positive Prediction	76 (TP)	06 (FP)
Negative Prediction	09 (FN)	77 (TN)

By processing the image tiles using the feature-extracting stages of MobileNetV2 offline, we generate a small train and validation data set containing 331 and 168 labeled vectors with 1280 dimensions (features). Although classifier training may proceed directly from here, training with a small number of training vectors usually leads to poor generalization, which is verified by Table I in the case of this data set. To solve this problem, data augmentation is applied to the training samples, as described in Section III. The binary classifier is then trained and validation performance is evaluated twice: for the basic 331/168 training/validation ratio, and for five folds, each with 399/100 training/validation ratio. The basic classifier is then applied, in raster-scan mode using non-overlapping tiles, to unannotated test images at 2592 × 1944 (from peripheral and thick blood smear slides) and 1596 × 1198 resolution (from a thick blood smear slide), and finally to the annotated test image tile data set from the thirteenth slide. Figure 2 summarizes the methodology.

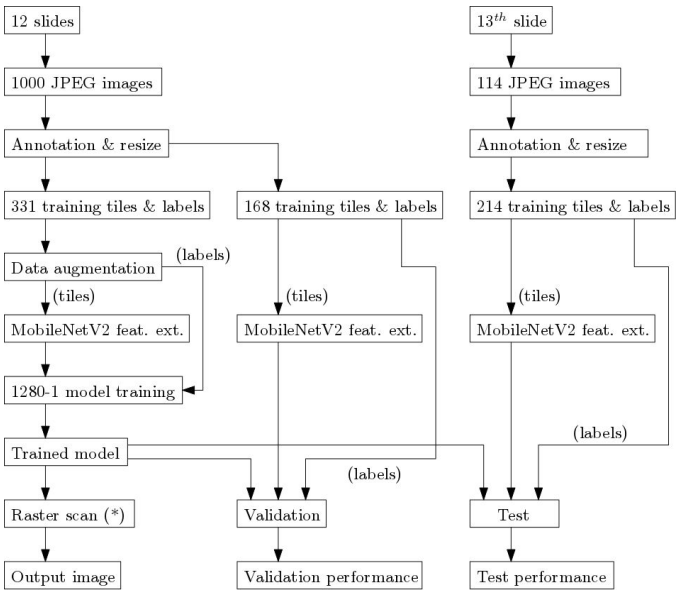


Figure 2. Data augmentation and the MobileNetV2 feature extraction are run offline. Feature extraction is pre-computed for the model training, validation and test, but raster scan (*) inference does include online MobileNetV2 feature extraction. To reduce clutter, raster scan input images are not shown.

IV. Results

A single-neuron classifier with 1280 inputs (features from MobileNetV2) is trained from scratch using Keras. The basic training process with 8275 training samples (which corresponds to 331 effective training samples that were augmented by a 25× factor) is illustrated by Figure 3. The validation data set contains 168 samples. The training configuration is as follows: 20 epochs, minibatch size equal to 32, dropout set to 0.2, learning rate set to 0.001 (i.e. the Adam optimizer is used, with its default learning rate). In the basic training process and also in the cross-validation experiments, the best validation results are obtained after approximately 10 epochs. A confusion matrix with validation results is shown in Table II and the respective accuracy, precision, sensitivity (i.e. recall), and F1-score values are shown in Table III. A few classification examples are shown in Figure 4.

To perform a cross-validation experiment with five folds, we first divided the dataset (first fold) into its first 399 samples, which were used for training, and its last 100 samples, which were used for validation (regardless of their originating image fields). In each of the four subsequent folds, the 100 validation sample positions were shifted by 100 units towards the first sample position in the dataset. In every fold, the same 25× data augmentation factor was applied to the 399 samples.

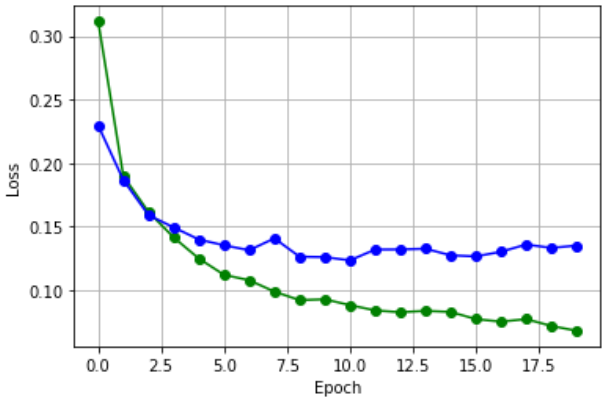


Figure 3. Training (green) and validation (blue) loss curves.

TABLE II. Basic training results. Confusion matrix obtained from validation data. TP and TN stand for true-positive and true- negative counts. FP and FN stand for false-positive and false-negative counts.

	Positive Target	**Negative Target**
Positive Prediction	83 (TP)	04 (FP)
Negative Prediction	02 (FN)	79 (TN)

TABLE III. Basis training results: precision, sensitivity, and F1-score values on the validation set.

Accuracy	(TP+TN)/168 = (83+79)/168	96.4%
Precision (P)	TP/(TP+FP) = 83/(83+4)	95.4%
Sensitivity (S)	TP/(TP+FN) = 83/(83+2)	97.6%
F1-score	2PS/(P+S)	96.5%

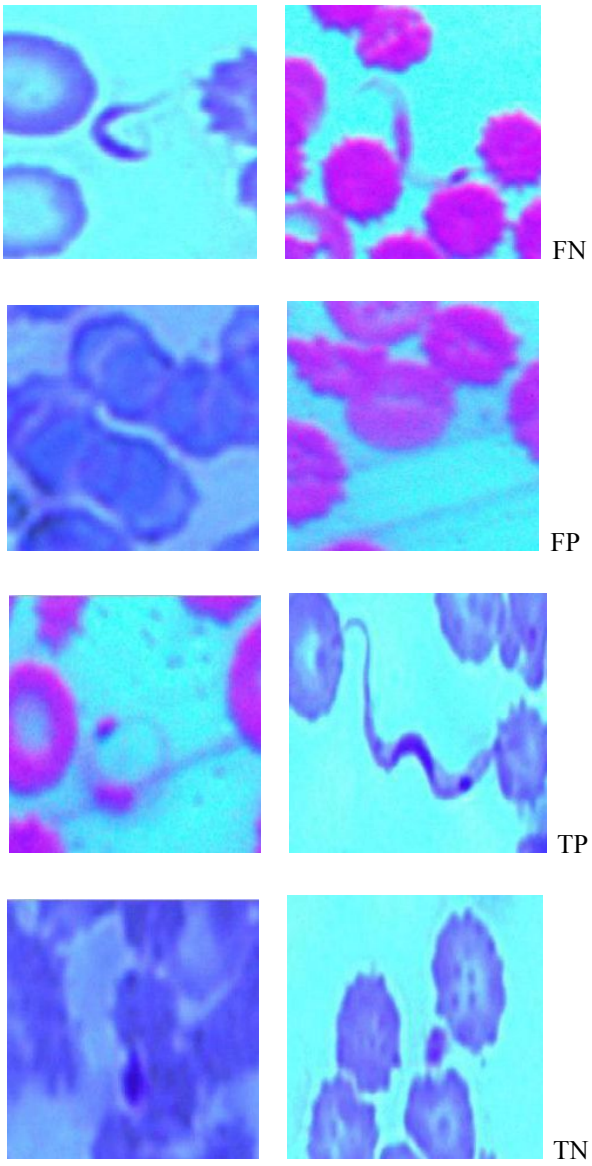


Figure 4. Validation result examples: from the top to the bottom, each row contains two samples that yielded false negative results, false positive results, true positive results, and true negative results.

TABLE IV. CROSS-VALIDATION EXPERIMENT RESULTS (FIVE FOLDS).

	P	S	F1	Accuracy
Fold 1	48/50	48/49	97.0%	97.0%
Fold 2	47/52	47/54	88.7%	88.0%
Fold 3	51/52	51/59	91.9%	91.0%
Fold 4	43/43	43/45	97.7%	98.0%
Fold 5	44/50	44/48	89.8%	90.0%
Average	**94.5%**	**91.7%**	**93.0%**	**92.8%**

The cross-validation experiment results are shown in Table IV. They indicate that, except for Fold 2 (which, upon visual validation loss curve inspection, seems to correspond to a "failed training", as shown in Figure 5. Training was actually carried out effectively but, apparently, on a loss function landscape that does not match the correct classification task), the figures of merit of a simple classifier based on a single neuron and 1280 features from MobileNetV2 are reasonably stable. Further verification regarding data inconsistency, to look for explanations for the problematic landscape, is under progress.

Based on the observation regarding result stability, we proceed towards the development of a very simple "raster-scan" loop that enables application of the basic classifier to test images with unlabeled square tiles (i.e. unannotated classifier input samples) and allows for visual inspection of the classification results. Three examples are shown in Figures 6 (peripheral blood smear test), 7, and 8 (thick blood smear test). Figure 6 indicates very good results (100% accuracy) on test samples coming from peripheral blood smear tests, which was expected, as the training and validation data set is also composed by peripheral blood smear tests, even though significant color and contrast variations may be there, as discussed in Section III (see, for example, Figure 1).

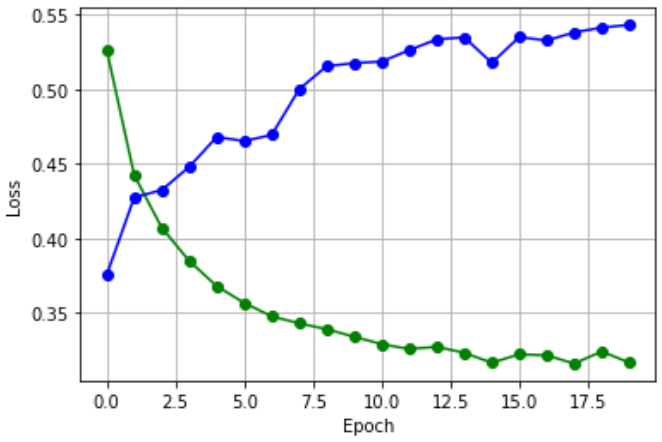


Figure 5. Fold 2 training (green) and validation (blue) loss curves.

Samples from thick blood smear samples, on the other hand, were not present in the training data set. As a consequence, as Figure 7 indicates, the basic classifier makes significant mistakes (two false-negative results and three false-positive results) on a thick blood smear test. According to the MobiletNetV2 input specifications, the square image tiles in Figures 6, 7, and 8 have size 224 ×224 ×3. It is possible that image resolution changes also play a role in the increased misclassification rate. To check that, we resize the image in Figure 7 to 2590 × 1944 and repeat the raster scan test. The results, shown in Figure 8, indicate improvements with respect to Figure 7: no more false-positive results, and three false-negative results that might be justified by misalignment between the parasite and the sliding window. Besides extending the training data set for diversity (i.e. for including

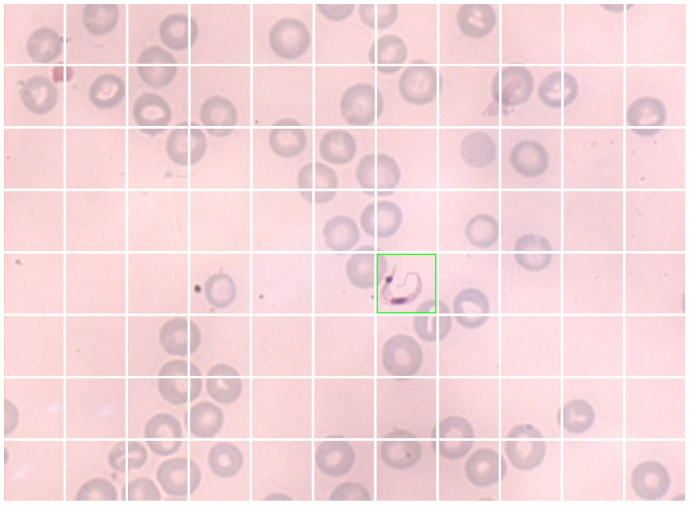


Figure 6. Raster scan test example. The tiled image corresponds to a 2590 × 1944 field that was manually extracted from a peripheral blood smear slide.

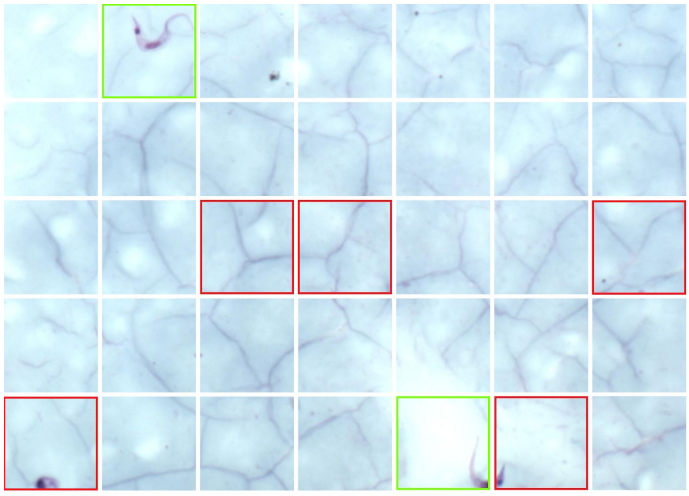


Figure 7. Raster scan test example. The tiled image corresponds to a 1596 × 1198 field that was manually extracted from a thick blood smear slide.

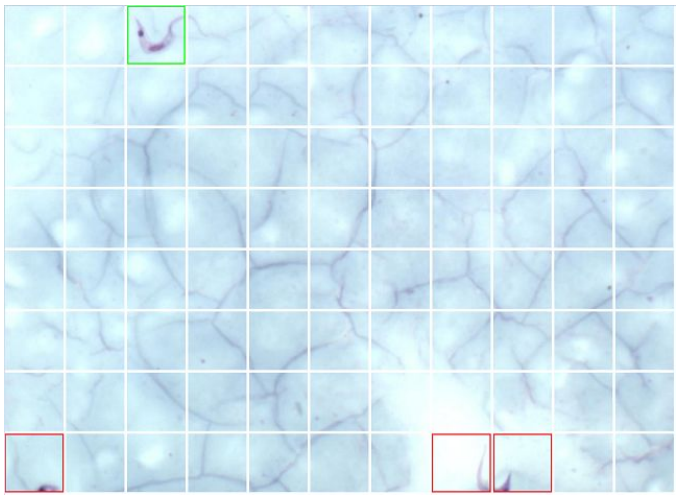


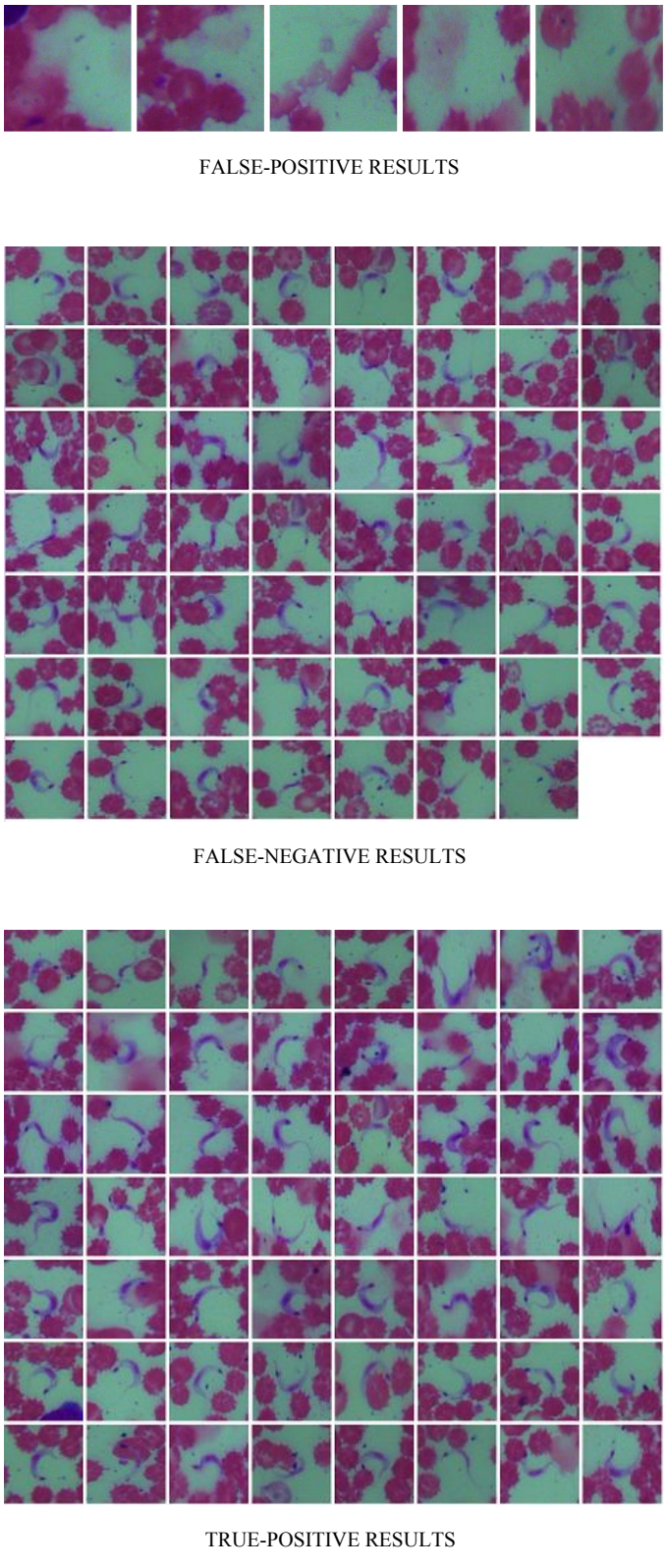
Figure 8. Raster scan test example. The tiled image corresponds to a 2590 × 1944 field that was manually extracted from a thick blood smear slide.

thick blood smear samples), the raster-scan system performance may also be improved by scanning the image fields with overlapping windows, and using post-processing positive count thresholds, at the cost of the corresponding additional computational resources, which is not taken into account in the present work.

Still, with respect to thick blood smear samples, we point out that *Trypanosoma cruzi* abounds throughout early infection days, and are thus easily found in microscopy exams with a single drop of fresh blood, which is the case of Figure 6. However, as the parasite count is considerably reduced at some point between six and eight weeks after infection, the thick blood smear technique becomes more convenient: by using a larger blood volume [2], it yields larger sensitivity than peripheral blood smear.

As a final test, we consider all images that are captured from a thirteenth blood smear slide. With respect to the train and validation twelve-slide data set, the images from this slide present significant variations in color, contrast, and overall aspect, which can be seen by comparing the square image titles in Figures 1 or 4, and Figures 9 or 10. To generate the manually annotated square image tiles, we follow the procedure that was outlined in Section III (Figure 2). The test results are shown in Figures 9 and 10. The test confusion matrix, which is shown in Table V, corresponds to 72.0% accuracy, 91.8% precision, 50.5% sensitivity, and 65.1% F1-score.

These test results indicate a large number of false negative results, for reasons that are not immediately clear from visual inspection of the image tiles in Figure 9, except for the aspect differences that were previously mentioned. In many of those tiles, the parasites are still clearly visible. It is possible that, by running this test on image tiles from an input image with



FALSE-POSITIVE RESULTS

FALSE-NEGATIVE RESULTS

TRUE-POSITIVE RESULTS

Figure 9. Square image tiles from images extracted from the thirteenth slide. They were manually annotated, and then used for classifier testing. All false positives (5) are shown at the top, all false negatives are shown in the middle (55), and all true positives are shown at the bottom (56).

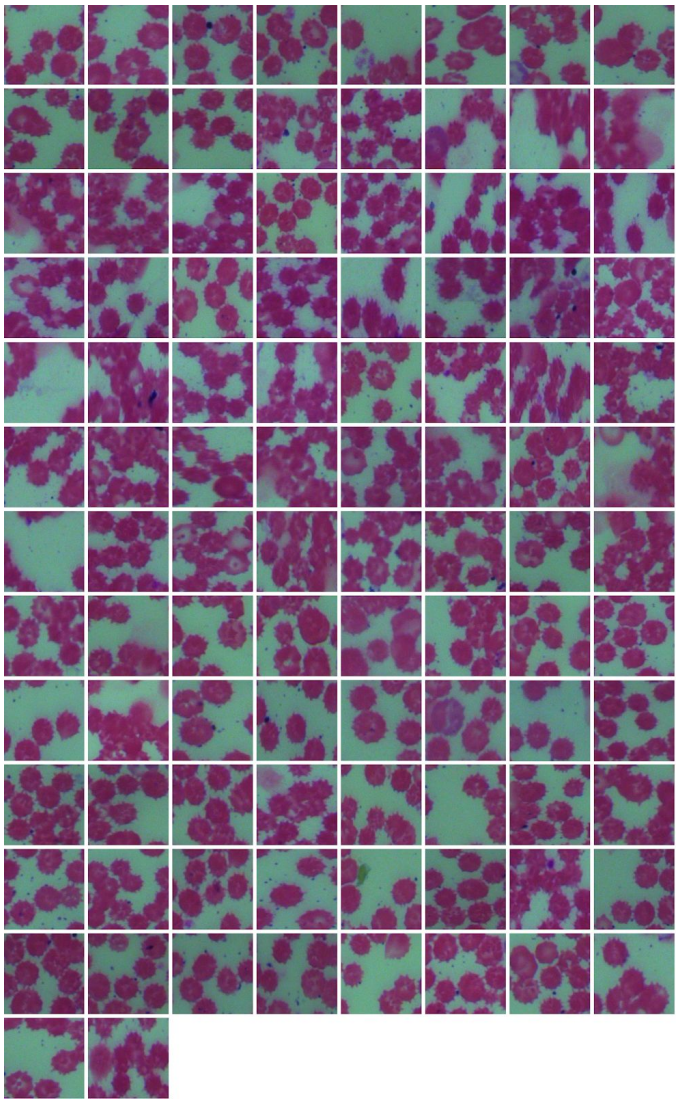


Figure 10. Square image tiles from images extracted from the thirteenth slide. They were manually annotated, and then used for classifier testing. This figure is a continuation of Figure 9. It shows all true negative results (98).

a different resolution, the results change somewhat. However, to ensure low false-negative rates (i.e. high sensitivity), we must next consider three points: i) an investigation on the features corresponding to the image tiles in the middle part of Figure 9, to see whether performance may still be improved with the currently available twelve-slide training and validation data set; ii) running additional tests with other new slides; iii) including additional samples in the training and validation data set, and reformulating the test sets to keep independence between train/validation and test samples; iv) test new classifier topologies, for example, with one or more fully connected hidden layers.

TABLE V. Confusion matrix obtained from the application of the basic classifier to all square image tiles from the thirteenth slide, as shown in Figures 9 and 10.

	Positive Target	**Negative Target**
Positive Prediction	56 (TP)	05 (FP)
Negative Prediction	55 (FN)	98 (TN)

In terms of execution time, we point out that inference is rather fast: raster scans such as the ones shown in Figures 6, 7, and 8 take less than five seconds to run on the entire input image. Taking into account that Figures 6 and 8 have 88 tiles, the time taken by the MobileNetV2 feature extraction and the subsequent binary classification is below 57 ms in the available computing infrastructure which, as described in Section II, is expected to be faster than a mobile phone core.

V. Conclusions

In this work, we used a MobileNetV2 feature extractor that had been previously pre-trained on ImageNet, and a fined-tuned fully-connected binary classifier consisting of a single neuron, to design an image analysis method for acute-phase Chagas disease diagnosis from blood smear samples. Accuracy values were 96.4% on the validation set and 72.0% on an independent test set. Although precision and sensitivity are below those reported in [3] (the image analysis methods based on the joint application of boosting and support-vector machines have been previously reported to yield accuracy values around 99%), up to our knowledge, the present paper is the first one to report an application of deep neural networks to Chagas disease diagnosis. As we seek to improve the performance of the currently proposed classifier, in future work, we will look into complexity comparisons with the boosting approach. Possible future improvements include: reducing false positive rates, statistical performance analysis using larger datasets, dataset improvement aiming at better neural network design, automated annotation methods, and raster-scan improvement based on overlapping tiles. We expect the improvements to eventually lead to a useful computer-aided tool for Chagas disease diagnosis.

Acknowledgments

The authors thank Professor Helena Keiko Toma and the Parasitic Disease Laboratory at FIOCRUZ for kindly providing some of the blood smear slides that were used in this work. They also thank the Neuroanatomy II Laboratory, at the Biophysics Institute, and the Inflammation and Immunity Laboratory, at the Microbiology Institute, for granting access to their microscopes. This work has been supported in part by CNPq under projects 432997/2018-0 and 309602/2016-5, and in part by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) Finance Code 001.

References

[1] L. S. Sangenito, V. S. Santos, C. M. d'Avila-Levy, M. H. Branquinha, A. L. S. Santos, and S. S. C. Oliveira, Leishmaniasis and Chagas Disease – neglected tropical diseases: treatment updates. *Current Topics in Medicinal Chemistry*, vol. 19, no. 3, 2019.

[2] L. Rey. *Parasitologia*. Rio de Janeiro: Ed. Guanabara-Koogan, Third Edition (in Portuguese), p. 39 and p. 49, 2010.

[3] V. Uc-Cetina, C. Brito-Loeza, and H. Ruiz-Piña, Chagas Parasite Detection in Blood Images Using AdaBoost. *Computational and Mathematical Methods in Medicine*, Hindawi, vol. 2015, article ID 139681, 2015. DOI:10.1155/2015/139681

[4] F. F. Norman and R. López-Vélez. Chagas Disease: Comments on the 2018 PAHO Guidelines for diagnosis and management. *Journal of Travel Medicine*, 2019. DOI:10.1093/jtm/taz060

[5] V. Uc-Cetina, C. Brito-Loeza, and H. Ruiz-Piña. Chagas parasites detection through Gaussian discriminant analysis. *Abstraction and Application*, vol. 8, pp. 6-17, 2013.

[6] R. Soberanis-Mukul, V. Uc-Cetina, C. Brito-Loeza, and H. Ruiz-Piña. An automatic algorithm for the detection of *Trypanosoma cruzi* parasites in blood sample images. *Computer Methods and Programs in Biomedicine*, vol. 112, no. 3, pp. 633-639, 2013.

[7] K. He, X. Zhang, S. Ren, and J. Sun, Delving Deep into Rectifiers: Surpassing human-level performance on Imagenet classification. *In Proc. 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026 - 1034, Santiago, Chile, December 2015.

[8] M. Poostchi, K. Silamut, R. J. Maude, S. Jaeger, and G. Thoma, Image analysis and machine learning for detecting malaria. *Translational Research*, vol. 194, pp. 36-55. DOI:10.1016/j.trsl.2017.12.004

[9] Rajaraman et al., Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. PeerJ 6:e4568, 2018. DOI:10.7717/peerj.4568

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks, *arXiv:1801.04381v4*, 2018.

[11] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[12] https://storage.googleapis.com/mobilenet_v2/checkpoints/mobilenet_v2_1.0_224.tgz and https://github.com/tensorflow/models/blob/master/research/slim/nets/mobilenet/README.md

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, December 2015. DOI:10.1007/s11263-015-0816-y

[14] http://keras.io

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization, *arXiv:1412.6980*, 2014.