THE ANNALS OF "DUNĂREA DE JOS" UNIVERSITY OF GALATI
FASCICLE III, 2020, VOL. 43, SPECIAL ISSUE, ISSN 2344-4738, ISSN-L 1221-454X
ELECTROTECHNICS, ELECTRONICS, AUTOMATIC CONTROL, INFORMATICS
Article DOI: https://doi.org/10.35219/eeaci.2020.2.02

CLASSIFYING SKIN MOLES USING CONVOLUTIONAL NEURAL NETWORKS

Adriana Stratulat-Diaconu[*], **Adina Cocu**[*]

[*]*Department of Computer and Information Technology,*
The University "Dunărea de Jos" of Galati, Romania
(e-mail: adriana.stratulatd@yahoo.com, adina.cocu@ugal.ro)

Abstract: The purpose of the paper was to develop an application that is capable to upload a picture and analyze it in order to determine melanoma lesions using artificial intelligence techniques. The proposed application is designed to use a previously trained convolutional neural network to recognize melanoma. For training, the examples from two known benchmarks were used and several attempts were made to find the best model driven by the neural network. The predictability rate is 0.95. The average time for obtaining the respond is 7 seconds.

Keywords: melanoma, mole, convolutional neural network (CNN), Artificial Intelligence.

1. INTRODUCTION

Skin cancer is quite prevalent worldwide and the early discovery of potentially dangerous lesions increases the chances of cure. Doctors recommend examination made by individuals each month and by specialist once a year. Each person can observe only the major changes that occur in skin pigmentation, often ignoring those lesions that are at risk to turn into melanoma. Melanoma is a sort of skin cancer that is growing around pigment producing skin cells and is forming around the moles. Hence the idea of developing an application that allows the loading of an image with a skin mole based on which to provide a classification of being or not melanoma.

The response of the application is calculated using a convolutional neural network developed in the Python programming language and trained using classified images provided by International Society for Digital Skin Imaging (ISIC) and University of Groningen.

Melanoma is a malignant tumor of the skin, of pigment origin, composed of melanin-containing cells. Both clinical and statistical research confirmed that melanoma is rapidly evolving, causing a rather high mortality among the population worldwide. Although it accounts for only 5% of all skin cancers, it accounts for over 75% of the deaths caused by these.

If this type of skin cancer is detected at an early stage, then it becomes treatable (Cancer.net). The first symptoms of melanoma are changes in the shape and color of moles that can be produce within weeks. Therefore, doctors recommend careful observation of pigmented skin lesions and when any changes are discovered it is recommended to visit a healthcare professional. Due to the need for early detection, computer algorithms have been developed to help in early computational diagnosis of typical skin lesions.

There are several applications available on the market that can detect different types of cancer, but many of them are commercially licensed. Most of them are made for mobile and offer some facilities like mole library, image management, reminders and graphs. The performance of these applications depends mainly on the quality and accuracy of the images that are linked by camera resolution. For training and detecting stages, the proposed solution uses images taken with high resolution cameras.

In the recent years, many studies have been conducted in the field of image classification. Pre-trained neural networks have been created and trained on certain types of images, such as GoogleNet, ResNet, AlexNet. Several studies have been directed in the field of identifying different skin abnormal cells from imaging. Thus, using the pre-trained AlexNet convolutional neural network, (Pomponiu and all, 2016) obtain an accuracy of 93.64% on a data set of 399 images. Another study, also conducted with the AlexNet network (Codella and all, 2015), used the ISIC set with 2624 images, with 93.1% accuracy. In another research, the convolutional neural network ResNet-152 (Mendes, da Silva, 2018) trained on 3797 images was tested on 956 images and obtained accuracies between 91% and 96% (for 12 types of cell lesions). A ResNet-50 convolutional NN (Matsunaga, 2017) was used to classify lesions from ISIC benchmark with 0.985 validation score. Different values of parameters were tested in GoogleNet incarnation of the Inception architecture (Szegedy and all, 2014), (Ridell and Spett, 2017).

All the mentioned researches used various image preprocessing techniques (data augmentation) or used different data sets for training and testing or they combined several artificial intelligence techniques (SVM vector machine support, K-nearest neighbor algorithm - KNN).

In this study, for image classification, the proposed technique uses neural network developed in Python. A convolutional neural network (ConvNet / CNN) is a deep learning algorithm. A CNN takes an input image, assign importance (learning difficulties and biases) to different objects in the image and differentiate between them.

This solution has been chosen because the pre-processing techniques required in a CNN are much lower compared to other classification algorithms. The CNN compensates the pre-processing stages with efficient training and parameters adjusting.

The hyperparameters necessary to define a convolutional neural network are the following:

- Number of epochs that defines the number of times when the learning algorithm will completely go through the entire set of training data. In each epoch the training set could update the internal parameters of the model. An epoch contains of one or more batches.
- Batch size that defines the number of samples to be worked on before updating the internal parameters of the model.
- Learning rate that controls how much we adjust the weights of our network with respect to the loss gradient.

2. DESCRIPTION OF THE SOLUTION

To generate the model, Teachable Machine, a free web-based tool offered by Google, was used. This tool is flexible and facilitates the creation of machine learning models using Tensorflow.js, a library for machine learning written in JavaScript, to train and run models made in the browser.

TensorFlow uses Keras API to design neural networks for machine learning models. This tool has a pre-trained neural network and added images become the last layer of the neural network. This tool generates a file with the extension .h5 proprietary TensorFlow data format. It was processed in Python and connected to an interface for ease of use. (Fabio Perez, 2019).

For training the model, there were used two data sets from ISDIS (International Society for Digital Skin Imaging) - "ISIC 2017: Skin Lesion Analysis Towards Melanoma Detection for Training" and from the University of Groningen, Department of Dermatology - "MED-NODE: A computer-assisted melanoma diagnosis system using non-dermoscopic images" (Giotisa at all, 2015). The proportion between images with and without melanoma in the two data sets are: 372/1628 and 70/100.

The Teachable Machine tool uses 85% of the provided images to train the model and 15% of the images for testing. The images were trained using a number of epochs between 50 and 4000, a batch size of 16 and a learning rate of 0.001. For each data set, 12 models were trained, and the first 4 models were chosen that have the highest accuracy first for melanoma and then for non-melanoma.

It is known there are different ways to improve the neural network performance as like: increase the number of neuron or hidden layers, change activation function, increase number of learning data sets, preprocess the data. The Teachable Machine tool offers only the possibility to change the CNN hyperparameters, not the structure of CNN or activation function. For this reason, in order to increase the accuracy, the concatenation of the two data sets will be tested.

The two data sets were entered separately in the tool to track the learning ability with a smaller number of training data. The only parameter changed during the training was the number of epochs. The accuracy result for the best four models corresponding to the two training datasets are presented in Tables 1 and 2, so it is possible to compare them.

Table 1 The training results ISIC data set

Melanoma images	**Non-melanoma images**	**Epochs**	**Accuracy Melanoma**	**Accuracy Non-melanoma**
372	1628	100	**0.43**	**0.92**
372	1628	2000	**0.39**	**0.91**
372	1628	3500	**0.39**	**0.85**
372	1628	2500	**0.38**	**0.93**

In Table 1 the data of the models trained using data from the ISIC set are presented. A large discrepancy can be seen between the small number of images with melanoma and the large number of non-melanoma images (372 versus 1628) that is projected in the accuracy result (under 0.43 and over 0.85). This comparison shows an over-learning for non-melanoma, so the results do not show a major improvement in the resulting accuracy as the changing of the number of epochs.

In Table 2 are presented accuracy results for the entire MED-NODE dataset, where the proportion between melanoma and non-melanoma images are more equilibrate (70 versus 100). Because of this, the obtained accuracy is close for the two classes (over 0.82 for both classes). In order to have more training data, the two data sets were concatenated (proportion melanoma vs. non-melanoma is 442/1728) and 12 models were trained, choosing the first four with the highest accuracy with priority for melanoma class.

Table 2 The training results MED-NODE data set

Melanoma images	**Non-melanoma images**	**Epochs**	**Accuracy Melanoma**	**Accuracy Non-melanoma**
70	100	3500	**1**	**0.93**
70	100	1000	**1**	**0.87**
70	100	350	**0.91**	**0.87**
70	100	1500	**0.82**	**0.93**

Table 3 The training results with the two combined datasets

Melanoma images	**Non-melanoma images**	**Epochs**	**Accuracy Melanoma**	**Accuracy Non-melanoma**
442	1728	4000	**0.36**	**0.93**
442	1728	3000	**0.34**	**0.91**
442	1728	2000	**0.32**	**0.9**
442	1728	1000	**0.31**	**0.88**

The concatenated data set from the two benchmarks presents a discrepancy of accuracy between the two classes. It can be observed that, when the number of cases learned from the two classes is balanced, their predictive accuracies are closed close (Table 2).

When the number of cases is disproportionate, then the accuracy of the classification of the two classes is also disproportionate (Tables 1 and 3). In conclusion, by increasing the training set throughout concatenation, the accuracy did not increase significantly (just with 0.01 for non-melanoma) and decreased by 0.08 for the melanoma class.

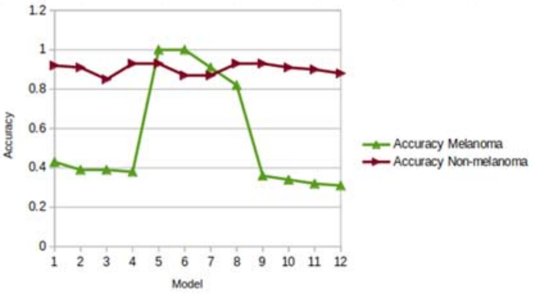


Fig.1. Accuracy comparison

From the graph presented in Figure 1, it can be seen that the first 4 samples are weaker (data from Table 1). The following 4 (data from Table 2) show that for both melanoma and non-melanoma, the graph increases considerably compared to the previous phase. The last 4 tests (Table 3) show an increase in non-melanoma values but comparable with the 4 models for melanoma.

Comparing the results, the model trained for 372 images with melanoma and 1628 without melanoma from ISIC data set, with 100 epochs, with the best calculated accuracy, 0.92 for melanoma and 0.43 for non-melanoma has been chosen.

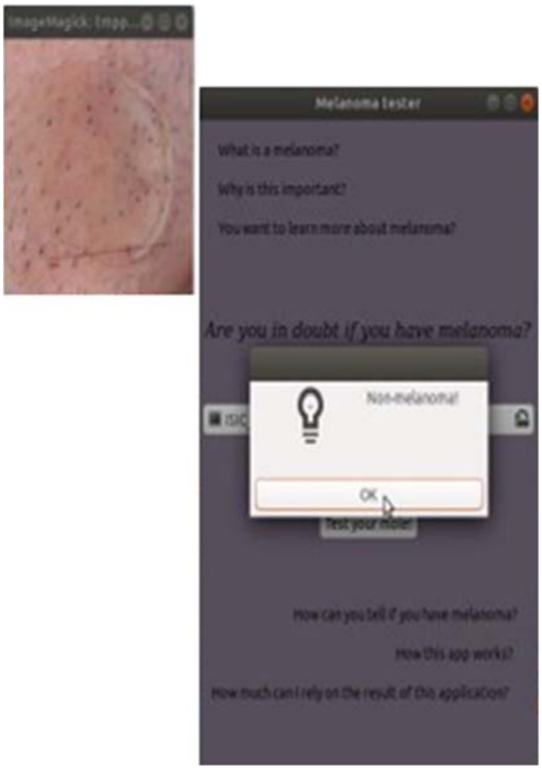
The Teachable Machine tool does not allow reusing the model if the browser session has been restarted but allows saving the model in a proprietary format. Hence the idea of creating an application that includes the trained model. The model chosen, in order to be used at any time, by any user without programming knowledge, was included in an application with graphical user interface. For validation, the developed application was used to create the confusion matrix.

The application in which the mole classification model was incorporated has a graphical interface created in Python. It provides information on what melanoma is, why testing is important and who you can contact in case of doubt. Also provides information about how this method works and about the accuracy of the prediction (Fig. 3).

The model was used so that when you choose an image it generates by clicking on the "Test your mole" button an answer of "Melanoma", "Non-melanoma" or "You should consult a doctor" (Fig. 2).

Specifically, when the user clicks the button, the model will load, creating the array of the appropriate shape to be inserted in the Keras template. The chosen image will be uploaded, resized to 224x224 and then transformed into a NumPy array and normalized (Fig. 4). A NumPy array is a central structure of the NumPy library (Numeric Python).

The prediction will be generated based on the model, and the final result will be issued based on the interval in which they are (Fig. 3).


ImageMagick: Imag...
Melanoma tester
What is a melanoma?
Why is this important?
You want to learn more about melanoma?
Are you in doubt if you have melanoma?
Non-melanoma!
OK
Test your mole!
How can you tell if you have melanoma?
How this app works?
How much can I rely on the result of this application?

```
value = self.neural_net()
if value[0][1] > value[0][0]:
  wx.MessageBox(‘Non-melanoma’)
elif value[0][0] > 0.4 and value[0][0] < 0.6:
  wx.MessageBox(‘You should consult a doctor!’)
else:
  wx.MessageBox(‘Melanoma’)
```

Fig.2. Setting application answers based on CNN output

Thus, if the value for melanoma is lower than that for non-melanoma it will return negative for melanoma, if the value of melanoma is between 0.4 and 0.6 a consultation will be indicated and otherwise it will return positive for melanoma.

Fig.3. Application interface

Fig.4. Using generated model from Teachable Machine into application with user interface

```
np.set_printoptions(suppress = True)
model = tensorflow.keras.models.load_model(‘PATH_TO_MODEL’)
data =   np.ndarray(shape = (1, 224, 224, 3), dtype = np.float32)
image = Image.open(self.file_open.GetPath(), mode = ‘r’)
size = (224, 224)
image = ImageOps.fit(image, size, Image.ANTIALIAS)
image_array = np.asarray(image)
image.show()
normalized_image_array = (image_array.astype(np.float32) / 127.0)–1
data[0] = normalized_image_array
prediction =  model.predict(data)
```

In predictive analytics, a table of confusion or confusion matrix is a table with two rows and two columns that reports the number of false positives, false negatives, true positives and true negatives.

The confusion matrix estimates the performance of the chosen model, so for 50 tests for 10 images with melanoma and 40 with non-melanoma. The results are presented in Table 4.

Table 4 Confusion matrix for chosen model

Actual Class		**Melanoma**	**Non-Melanoma**
Predicted class	Melanoma	9 True Positives	0 False Positives
	Non-Melanoma	1 False Negatives	40 True Negatives

The used images were not chosen from the set for training and testing in the model learning phase. The results are encouraging because it is a single image with melanoma that was classified as being without melanoma and 50 images tests were performed.

3. CONCLUSIONS AND FUTURE DEVELOPMENT

The proposed method uses neural networks to separate melanoma from the healthy moles. The classification was made correctly for all welded images in the specified image bases. As a result, the application can be used successfully in separating melanomas from healthy nevus (with the mentioned accuracy).

In the future, this algorithm could be implemented in an application accessible to any device.

4. REFERENCES

Codella N., Cai J., Abedini M., Garnavi R., Halpern A., Smith J.R., (2015), *Deep learning, sparse coding, and SVM for melanoma recognition in dermoscopy images*; Proceedings of the 6th International Workshop on Machine Learning in Medical Imaging; Munich, Germany. pp. 118–126. https://link.springer.com/chapter/10.1007/978-3-319-24888-2_15.

Dermatology database used in MED-NODE, (2015), http://www.cs.rug.nl/~imaging/databases/melanoma_naevi/.

Fabio Perez, (2019), *Data Augmentation for Skin Lesion Analysi*s - Best Paper Award at the ISIC Skin Image Analysis Workshop @ MICCAI, https://github.com/fabioperez/skin-data-augmentation.

Giotis I., Molders N., Land S., Biehl M., Jonkman M.F., Petkov N., (2015), *MED-NODE: A computer-assisted melanoma diagnosis system using non-dermoscopic images*.

Ioannis Giotisa, Nynke Moldersb, Sander Landc, Michael Biehla, Marcel F.Jonkmanb, Nicolai Petkov, (2015), *MED-NODE: A computer-assisted melanoma diagnosis system using non-dermoscopic images*, Ed. Elsevier, Expert Systems with Applications, Volume 42, Issue 19.

Keras for Beginners: Building Your First Neural Network, (2019), https://victorzhou.com/blog/keras-neural-network-tutorial/.

Matsunaga K., Hamada A., Minagawa A., Koga H., (2017), *Image Classification of Melanoma, Nevus and Seborrheic Keratosis by Deep Neural Network Ensemble*. https://arxiv.org/ftp/arxiv/papers/1703/1703.03108.pdf.

Mendes D.B., da Silva N.C., (2018), Skin lesions classification using convolutional neural networks. arXiv. 20181812.02316, https://arxiv.org/pdf/1812.02316.pdf.

Pomponiu V., Nejati H., Cheung N.M. Deepmole, (2016), *Deep neural networks for skin mole lesion classification*; Proceedings of the IEEE International Conference on Image Processing (ICIP); Phoenix, AZ, USA. 25–28 September 2016. https://ieeexplore.ieee.org/abstract/document/7532834.

Ridell P., Spett H., (2017), *Training Set Size for Skin Cancer Classification Using Google's Inception v3,* Degree Project Computer Engineering, First Cycle, Stockholm, Sweden, https://www.diva-portal.org/smash/get/diva2:1112097/FULLTEXT01.pdf.

Rosebrock Adrian, (2014), *Skin Detection: A Step-by-Step Example using Python and OpenCV*, https://www.pyimagesearch.com/2014/08/18/skin-detection-step-step-example-using-python-opencv/?fbclid=IwAR31X2qfSvSFP8K49J8Eszv3n29P0uNmcvYJIQs-n0_g8Gyafivd0h64tnQ.

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A., (2014), *Going deeper with convolutions*, arXiv. https://arxiv.org/pdf/1409.4842.

*Teachable Machine Library – Image, (*2020), https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/image.