Predicting patient age using deep neural network image classification of dental x-rays

Declan Fitzgibbon[1], Reynaldo Quintero[1], and Wenhan Yang[1]

[1]MIAGE Innovative Information Systems, University Toulouse 1 Capitole, France

Abstract

In recent years we have seen massive advancements in the field of medical image processing. With similar leaps in the areas of artificial intelligence and big data, we can now, more than ever depend on these technologies accomplish detailed analysis for the betterment of science and patients alike. In this paper we examine a variety of deep neural network architectures for the purpose of image classification. We present the methodology and results of artificial neural network (NN) architectures designed to identify the age of patients using x-ray images of their teeth. We propose a final architecture, achieved in our experimentation and via the use of suggestions found in previous papers on this subject, that achieves a 1.73 training loss and 3.05 validation loss. This architecture includes the use of convoluted layers, max pooling, batch normalization and a gradient descent optimizer. We finally recommend the areas of the architecture that can be further optimized.

Index terms: neural network, classification, medical image processing

1 Introduction

The ultimate goal of developing medical imaging classification tools and architectures is to benefit patients in their care and treatment. The quality and effectiveness of the tools we develop directly impact the lives and well-being of those receiving medical care. The current state of medical image classification methods often puts it on par with human experts. [1] Hence, it is not inconceivable that by furthering advancements in this field, we may surpass human capability in medical image classification. The purpose of this paper is in this vein. The main operations of medical image processing are:

- X-Rays
- Ultrasonic
- Computed tomography (CT)
- Magnetic resonance imaging (MRI)
- Nuclear medicine imaging (NM) [2]

Since 2009, when Shi et al. wrote their survey of NNs for medical image processing, we have seen the advent of deep neural networks and the impact they have had, outperforming all other neural network architectures in this field. A particularly big step in the use of CNNs came in 2012 following the Imagenet Scale Visual Recognition Challenge (ILSVRC) in which Krizhevsky et al.'s won with a CNN with an error rate of 15%. This team went on to develop the Rectified Linear Unit (RELU) which we will explore latter in this paper.[3]

2 Material and methods

2.1 Datasets inspection

The data-set considered in this paper consists of x-ray images of patients teeth. 3327 patients were x-rayed to obtain this data. Various measurements were considered when devising the approach to the neural network design.

Measurements of the following teeth were taken from the lower left side of patients' mouth:

- Molar 2 (M2)
- Molar 1 (M1)
- Premolar 1 (PM2)
- Premolar 2 (PM1)
- Canine (C)
- Incisor 2 (I2)
- Incisor 1 (I1)

If any tooth was absent, it was replaced by the corresponding tooth in the lower right jaw.

Additionally, if wisdom were present in any region of the mouth they were accounted for in the data-set. The gender and age of the patient was also recorded.

```
<seaborn.axisgrid.FacetGrid at 0x7f069756aa58>
```

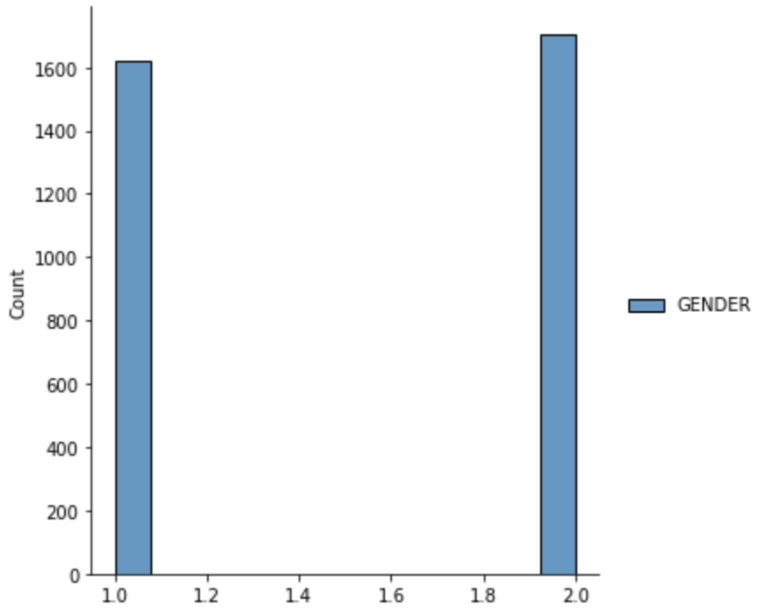

FIGURE 1: Gender distribution

The distribution of gender in the datasets are approximately even. Each gender contains around 1600 lines of data. It is balanced and considered to be well distributed to train the model.

```
<seaborn.axisgrid.FacetGrid at 0x7f067f649f28>
```

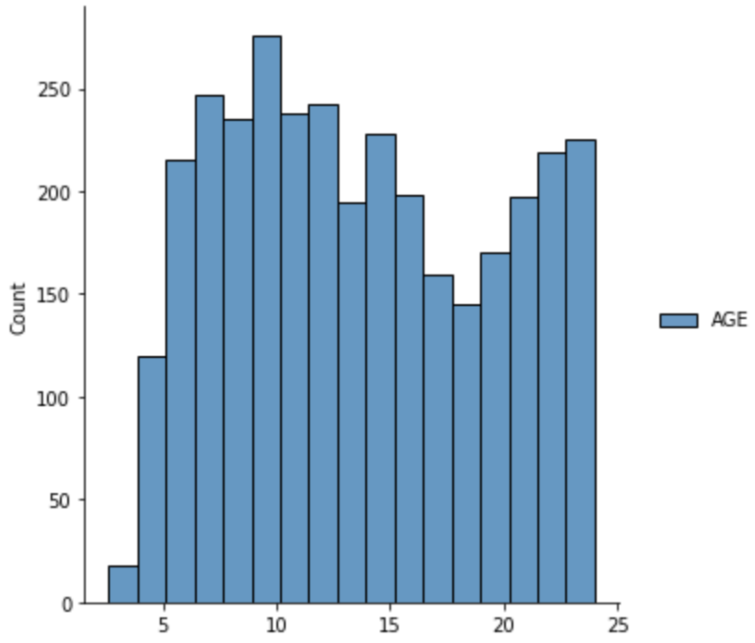

FIGURE 2: Age distribution

Across the distribution of age in the data-sets, the oldest patients are younger than 25 years, while the youngest subjects are around 2 years old. All ages, except the youngest patients of 2 year, have more than 100 lines of data. The distribution of age is approximately even. However, this indicates that the resulting model will not be exposed to people that are outside of this age range.

2.2 Data set treatment

Regrading data treatment, the two steps consist of preparing the input data-sets and training CNN model. At first in the data-sets, each patient has 12 parameters as an input to the CNN comprising of 11 teeth and a gender. Each tooth is represented in a single x-ray image. See below for examples.

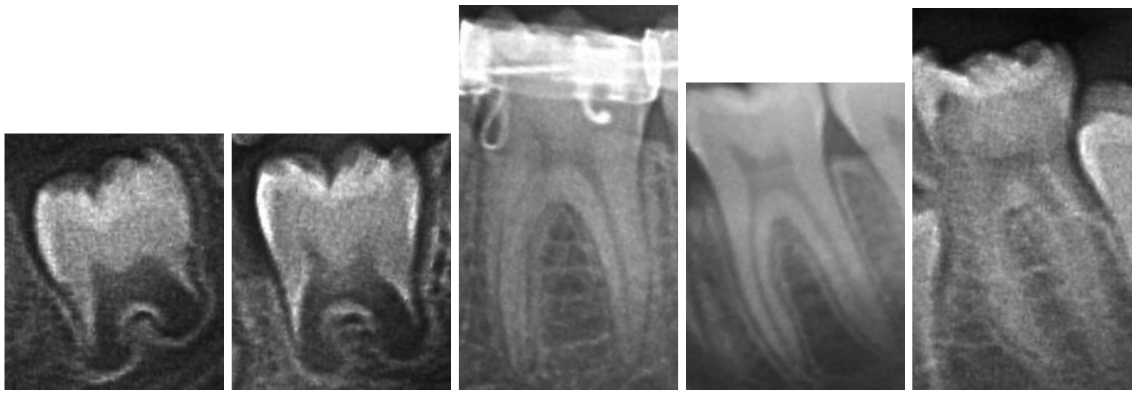
FIGURE 3: Dental x-rays

As per fig.3 it is clear that the images vary in size. In order to make the images uniform, every image is resized to 40×30. Gender is represented as a Numpy array in form of $30 \times 40 \times 3$ and combined with the other images. The gender which is represented by (1) is filled with 0 in Numpy array and gender (2) is filled with 255 in the Numpy array. For the missing images, they are represented as a Numpy array in form of $30 \times 40 \times 3$ filled with -255. The final step is to combine all 12 arrays as 3×4 and get a 120×120 array as a result.

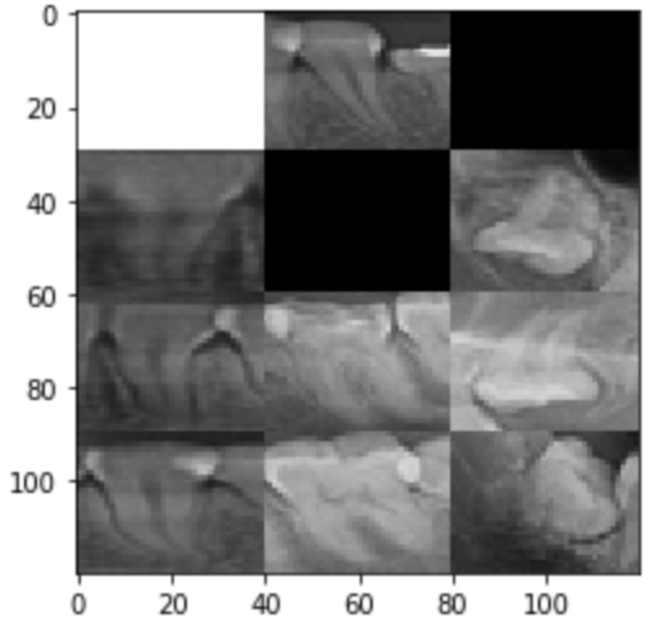

FIGURE 4: Image input

Fig.4 illustrates how the input array looks in photo format. The top left square represents the gender where white is gender (1) in the data sets, black is (2). The remaining images are teeth x-rays. Black images throughout the remainder of the array represent missing images in the data sets.

In order to train the model, the data is split as 75% training data sets and 25% testing data sets. Due to the limits of the resources, only 1000 lines of data is used to train the model.

As part of the experimentation, a different data set was implemented in which only the teeth that seemed empirically relevant were used. These were: the incisor teeth as they change form as the person ages (starting saw-shaped and progressively getting smoother over the years); molars since the presence of amalgam may be there, which is something that appears more commonly in more adults compared on children; and the wisdom teeth since their presence and position can be indicative of age. The sex of the person was also included, since some studies indicate tooth size can vary between genders [7].

In this case, the input is treated as described above, however, the final shape of the picture becomes 3×3 instead of 4×3.

2.3 Architectures

After the data-sets are prepared, the following step is to show different convolution neural network architectures and compare the results. All the networks use regression strategy, since the result that is being sought is a real number and not a classification between limited options.

The first architecture is based on the general architecture of CNN which includes 3 convolution layers with filters 32, 64, 128 respectively, ReLu activation function, kernel size of 3×3, Batch Normalization layer and dropout layer. Next is a flatten layer and 1 dense layer with 1024 weights and ReLu activation function. The last layer is a dense layer with 1 weight. The model has 118 million parameters in the end. The optimizer is Adam and loss is mean square error. This model is trained over 100 epochs and a batch size of 32.

2.3.1 Layers

In the concept of deep neural networks, the more layers added in the network, the more complex the mathematical function that can be modelled [4]. Thus, the new model has more convolution layers at each size of filter and more dense layers to test the performance.

2.3.2 Activation functions

Another important parameter in NN is activation function. In the previous model, the activation function is ReLu in every layer. ReLu is widely used in CNN [5] and is capable of speeding up the training process. LeakyReLu is another activation function based on ReLu but the only difference is that LeakyReLu does not output all of the negative values as 0. Instead, the leak helps to increase the range of the ReLU function[5]. However in these data-sets, the only negative input is -255 which is represented by the missing photos.

Since the target is age and linear regression output is the same as input [6] so the last dense layer is a linear dense layer with 1 neuron. For the remaining dense layers, the activation functions are Tanh. The range of the tanh function ranges from (-1 to 1) and it is also sigmoidal. The advantage is that the negative and positive inputs will be mapped strongly negative and positive and the zero inputs will be mapped near zero in the tanh graph [5]. It aggregates outputs between -1 and 1 which stabilises training sessions.

2.3.3 Training parameters

In these data-sets, generally, as the training epochs progress the better result is obtained but with massively diminished returns after 100 epochs. Therefore, every model is trained with 100 epochs and it should approximately obtain the best result possible. Given that the number of data points used is 1000, the batch size is set up to 32, so the model has 24 times back-propagation. The concept in this is to train the model as fast as possible with reasonable timing of back-propagation.

2.3.4 Gradient descent optimizer

Another area to be considered is the gradient descent optimizer that will be used alongside the model. The most used and documented optimizers [8] are: Stochastic gradient descent (SGD), SGD with Nesterov accelerated gradient, Adagard, Adadelta, RMSprop, Adam, AdaMax, Nadam and AMSGrad. These will consequently be tested so as to determine the best fit for the model produced, using the parameters suggested by Ruder [8].

2.3.5 Dropout layer

The purpose of dropout layers inside of a convoluted neural network is to avoid overfitting the neural network, by ignoring the input of a random set of neurons dictated by the percentage amount of the dropout specified [9]. The perfect amount of dropout has been discussed in different papers, but the general rule proposed is that a 50% amount is ideal [10]. We will, however, test a different range of values and select the amount that is better fitted. The values to be tested are: 25%, 35%, 45%, 50%, 55%, 65% and 75%.

2.3.6 Kernel size

Kernel size is a parameter that allows the CNNs that are used in image analysis, to study to different specificity the features from the pictures it's being fed. The size of this kernel is a parameter that is often defined through brute force testing, and in this case this will be done using some standard values [11]: 1×1, 3×3, 5×5 and 9×9.

3 Results

In the first general CNN model, the validation loss is around 80. After adding more layers, the model is 3 times as large as before and the parameters increase to 254 million. The validation loss is much improved and decreases to 18. However training such a large network is not efficient. And given the limited data, it is hard for such a large model to train a complex mathematics functions. Finally, the size of model is reduced as to only have 1 more convolution layer at each filter size and 1 more dense layer and the validation loss slightly increased.

From the analysis of the data-sets, LeakyReLu does not improve the validation loss better than ReLu. However, the following model will use LeakyReLu as activation fuction. On the other hand, the two activation functions of dense layers improved the validation loss from 18 to 4 and the validation loss is more stable than using ReLu.

Gradient optimizer test results:

Gradient descent optimizer:	Stochastic gradient descent (SGD)	SGD with Nesterov accelerated gradient	Adagard	Adadelta	RMSprop	Adam	AdaMax	Nadam
Loss:	3.22	3.35	10.62	8.38	5.31	5.24	4.20	3.51

There are also two combinations of data-sets that are tested. One is a full data-set with all the teeth and another one contains only sex and aforementioned important teeth (section 2.2). The validation loss of the full data-set obtains 3.69 and the other obtains a score of 4.33.

Dropout value results:

Dropout Value:	25%	35%	45%	50%	55%	65%	75%
Loss:	3.41	3.28	3.81	3.62	3.83	5.24	6.01

The loss does not differ greatly as we can see. As per the recommendations of the papers [10] and to avoid over-fitting, 50% is taken to train the rest of models.

Kernel size validation loss:

Kernel size:	1×1	3×3	5×5	9×9
Validation loss:	6.55	3.05	3.24	4.06

Finally, the best model achieved a 1.73 training loss and 3.05 validation loss, using the full data set. This model has 6 convolution layers among which are 2 convolution layers for each size of filter and kernel size is 3×3 and LeakyReLU activation function. Batch normalization layer and dropout layer with 50% drop following each convolution layer. 3 max pooling layers follow every 2 convolution layers. After the last convolution layer is flattened, 2 dense layers with 8192 and 512 neurons respectively and Tanh activation function are placed. The last layer is a dense layer with 1 neuron and linear activation function. The optimizer that showed the best result was the SGD Optimizer with a momentum of 0.9 and the usage of Nesterov accelerated gradient.

```
Epoch 95/100
24/24 [==============================] - 3s 124ms/step - loss: 1.6930 - val_loss: 3.3935
Epoch 96/100
24/24 [==============================] - 3s 124ms/step - loss: 1.6074 - val_loss: 3.6408
Epoch 97/100
24/24 [==============================] - 3s 123ms/step - loss: 1.6747 - val_loss: 3.5038
Epoch 98/100
24/24 [==============================] - 3s 124ms/step - loss: 1.7109 - val_loss: 3.3975
Epoch 99/100
24/24 [==============================] - 3s 124ms/step - loss: 1.5707 - val_loss: 3.3402
Epoch 100/100
24/24 [==============================] - 3s 124ms/step - loss: 1.7380 - val_loss: 3.0526
```

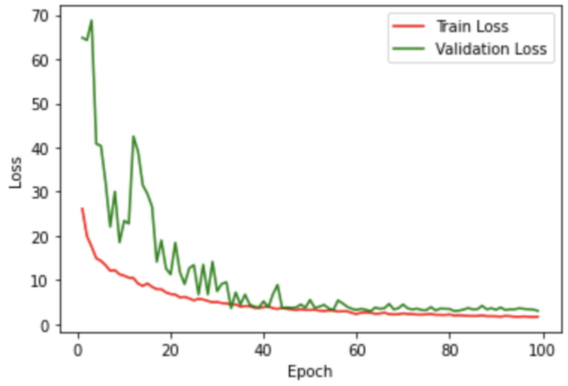

FIGURE 5: Image input

4 Conclusion and perspectives

The objective of this paper was to provide a model that was effective in predicting the age of a specific person given the X-Ray profile of their denture. For this, a data set was given that included a set of imaging from different teeth of a patient alongside with the sex and age of the patient.

Different architectures and configurations were tested, through testing of different suggestions provided in previous papers and investigations conducted. The specific areas tested were: amount and nature of the neural network layers, activation functions, gradient descent optimizers, amount of dropout in the dropout layer and kernel size. The best model achieved a 1.73 training loss and 3.05 validation loss using the full data set. In the experiment, the two changes of architecture that dramatically decreased the loss were: adding more layers and changing the activation function. After loss decreased to 4, it is very hard to improve with the limited resources we would have at this point.

As for recommendations, we would call for further experimentation with the optimization of the parameters for the optimizer chosen, as well as the further study of the parameters of the pooling layers, and the LeakyRelU function. Using all 3327 data points and training with more epochs could additionally give rise to optimisation of the existing model. Finally, we would also recommend utilizing a data-set that has a broader spectrum of ages.

References

[1] Samir S. Yadav and Shivajirao M. Jadhav (2019) *Deep convolutional neural network based medical image classification for disease diagnosis*, https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0276-2

[2] Zhenghao Shi, Lifeng He, Kenji Suzuki, Tsuyoshi Nakamura and Hidenori Itoh1 (2009) *Survey on Neural Networks Used for Medical Image Processing*https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4699299/

[3] Justin Kerlipo, Wang Jai Rao and Tchoyoson Lim (2018), *Deep Learning Applications in Medical Image Analysis*, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4699299/

[4] Assaad MOAWAD (2019) *Dense layers explained in a simple way*, https://medium.com/datathings/dense-layers-explained-in-a-simple-way-62fe1db0ed75

[5] Sagar SHARMA (2017) *Activation Functions in Neural Networks*, https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

[6] Assaad MOAWAD (2019) *Linear layers explained in a simple way*, https://medium.com/datathings/linear-layers-explained-in-a-simple-way-2319a9c2d1aa

[7] Eva Man YEE LEUNG et al. (2018) *A Comparative Analysis of Tooth Size Discrepancy between Male and Female Subjects Presenting with a Class I Malocclusion*, https://www.hindawi.com/journals/tswj/2018/7641908/

[8] Sebastian RUDER (2016) *An overview of gradient descent optimization algorithms*, https://ruder.io/optimizing-gradient-descent/index.html#gradientdescentoptimizationalgorithms

[9] Pierre BALDI and Peter SADOWSKI (2013) *Understanding Dropout*, https://proceedings.neurips.cc/paper/2013/file/71f6278d140af599e06ad9bf1ba03cb0-Paper.pdf

[10] Geoffrey HINTON et al. (2014) *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf

[11] Mark COLETTI et al. (2019) *Evolving larger convolutional layer kernel sizes for a settlement detection deep-learner on Summit*, https://www.osti.gov/servlets/purl/1631245