Predicting Bone Age Prediction Exploiting Images Through CNN Models

Suleyman Erim†

***Abstract*—This study investigates the use of convolutional neural networks (CNNs) for estimating the age of children based on hand bone radiographs. Five different base models were tested, including basic CNNs, inception blocks, residual skip connections, pre-activation skip connections, and inception blocks with skip connections. The models were trained with various hyperparameter choices, such as batch normalization, gender information, drop-out regularization, and data augmentation. The combination of inception blocks and batch normalization with gender information showed the best results in terms of accuracy. The mean absolute error between predicted age and the true age in months could be further decreased by gathering more information and using an ensemble approach. The study provides valuable experience and knowledge in the development of deep learning models and highlights the importance of the validation set during hyperparameter selection.**

***Index Terms*—Convolutional Neural Networks (CNN), Inception, ResNet, Batch Normalization, Age Prediction**

I. Introduction

The growth in computational power and the advancement of machine learning models has led to the widespread use of deep learning in computer vision across various fields such as health, manufacturing, and security. The ImageNet competition has played a significant role in boosting the performance of deep learning and convolutional models by providing them with ample data to learn from. This success should be leveraged to benefit the healthcare industry, where machines can perform repetitive tasks with greater accuracy and efficiency, freeing up specialists to focus on more complex challenges.

The RSNA pediatric bone age machine learning challenge represents an opportunity to enhance the use of automatic computer vision systems in healthcare by exploiting the feature extraction capabilities of deep learning models. The challenge requires participants to build a deep-learning convolutional neural network that can accurately predict the skeletal maturity of children from radiographs taken at two different hospitals [1].

Deep learning is a type of machine learning that uses multiple levels of feature representation and automatic feature extraction, achieved through the use of multiple layers of nodes in a network. The labeled images are processed through the network, and their features are represented as tensors that help the computer understand the task at hand. Convolutional neural networks are widely used in tasks such as facial recognition, image classification, and object detection [2].

The aim of this paper is to investigate the potential of deep learning in healthcare by comparing various state-of-the-art methods improved through the ImageNet competition. The manual assessment of skeletal bone age is a time-consuming process for specialists, and automating this task through deep learning can optimize their time and allow them to focus on more critical tasks.

II. Related Work

The advent of deep learning in image processing and computer applications was made possible by automatic feature extraction and reduced processing effort. One key development was the introduction of Convolutional Neural Networks (CNNs), which consist of two parts: the feature extraction part made up of a series of convolutional layers and the classification part made up of a series of fully connected layers.

In 1998, LeNet-5, a pioneering CNN, was introduced. Although it performed well on simple images, there was a need to improve it for high-quality images [3]. AlexNet improved on LeNet-5 by creating deeper networks to exploit image features and preventing over-fitting through techniques such as data augmentation and dropout as well as overcoming vanishing gradient problem with newly introduced ReLu activation function [4].

Data augmentation increases the size of the training data by using label-preserving transformations, such as image rotation, flipping and scaling, to avoid over-fitting [2]. Dropout sets the output of some neurons to zero with a probability of 0.5 during each iteration, allowing the network to learn more robust features [4].

VGGNet used smaller 3x3 kernels to extract finer-level features compared to AlexNet's larger kernels (11x11 and 5x5). The idea was that multiple stacked smaller kernels were better than a larger kernel because they increased the depth of the network and learning of more complex features and allowed for usage of more activation functions and the [5].

After VGG's idea to decrease number of decision parameters and continue with fixes kernel size, Inception block was introduced by Google. Inception module consists of different kernel size and max pool layer as a block to exploit differen features of the image at once rather than stacking layers on top of each other as in classical architectures [6].

The increased depth of networks has caused the problem of vanishing gradients, where the signal to change weights in

†Department of Mathematics, University of Padova,
email: sueleyman.erim@studenti.unipd.it

earlier layers becomes weak. When backpropagating error gradients, the signal decreases quickly and reaches zero, making it difficult for earlier layers to learn. He et al. addressed this by introducing skip connections, which provide a direct path for the gradient to reach earlier layers. This also helps the model learn an identity function, making it perform as well as earlier layers. With this technique, deep networks with 50, 101, or 152 layers can be trained, with lower complexity compared to smaller networks like VGGNet with 19 layers [7].

Initially, in skip connections, input was fed to residual blocks and the features were added to the input before feeding to the activation function. Later studies showed that manipulations on the shortcuts like scaling, gating, 1x1 convolutions, and dropout can reduce information flow and cause optimization issues. To address this, pre-activated residual blocks were introduced, which preserve the shortcut signal without any manipulation [8]. Additionally, Res-Net heavily uses batch normalization, which helps to handle saturating nonlinearities and internal covariate shift problems in early layers, and allows the network to use higher learning rates with less concern for initialization as the mini batches are normalized, scaled and shifted, reducing the risk of exploding or vanishing gradients [9].

III. Processing Pipeline

The data set consists of 14236 hand radiographs. The train, validation and test dataset is already splitted. The general processing pipeline is indicated in Fig. 1

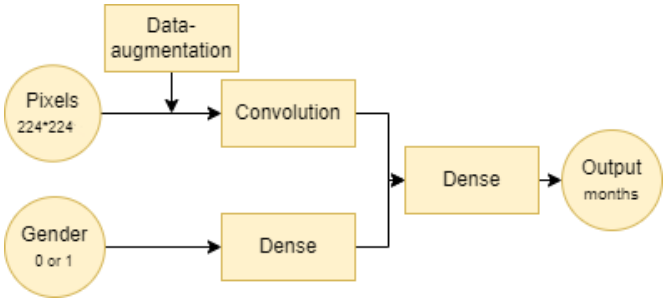

Fig. 1: Model Layout

First of all, the images, gender and age information gathered from files and training, validation and test datasets were created. While creating datasets preprocessing is applied to all of them, but augmentation is applied only on traning daraset. Data augmentation is optionally applied on training dataset for different models to experiment effect of data augmentation. Gender information is fed to network after passing a dense layer then concetenated with image features which are passed through convolutional layers. Different models were experimented with by incorporating and removing various techniques such as vgg layers, inception blocks, original and pre-activated residual blocks, drop-out, and batch normalization to assess their impact.

Inception block is indicated in Fig. 2. To reduce number of parameters and fasten training, bottleneck 1x1 kernel added before 3x3 and 5x5 filters and after max pooling layer.

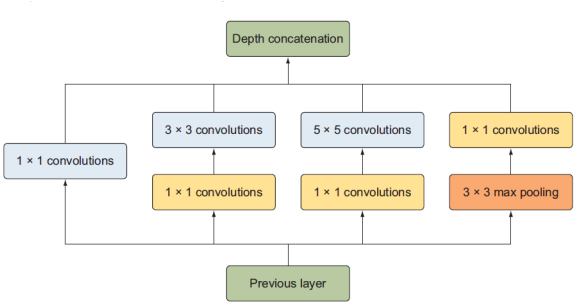

Fig. 2: Dimension Reducted Inception Block

A residual block with 3 convolutional layers and batch normalization is indicated in Fig. 3

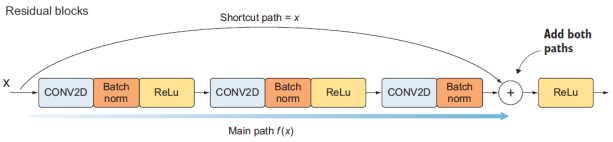

Fig. 3: Residual Block

The models consists of convolutional layers to extract features from images, optional 1 dense layer of gender information, then addition of image features and gender information, then additional dense layers and final 1 node dense layer to create a regression task. The prediction of age in months is regression problem with mean absolute error. To decrease training time and to handle local optimum early stopping and adaptive learning rate utilized.

IV. Signals and Features

The data was already splitted as training, validation and test set (12611 of images in training set, 1425 of images in validation set, 200 of images in test set). Since the ratio of training and validation set is between 10-20%, I have not change the number of training and validation samples. Images have different sizes, after trial and error for memory occupation, I have decided to resize images to 224*224 pixel with gray-scale. After resizing, the values of pixels standardized between 0-1 to fasten training process and also to handle saturation of gradients.

The age (in months) and gender (male : True) information was gathered from csv files and matched with images by ID's of images. Also gender information is binary encoded (male: 1, female: 0). Gender information is used for some models to decide if gender information increases models' capabilities to predict age in months. In addition, data augmentation was performed using rotation range of 20 degrees, horizontal/vertical translation up to 20%, random zoom up to 20% and a horizontal flip. This was done to force the network to learn features which are intrinsic to the patient rather than the imaging technique. In addition the training dataset repeated two times to increase number of examples fed to model.

V. Learning Framework

- **Input Layer:** The 3D tensor input is gray-scaled and resized to shape (224*224).Since the images are already black and white, I gray scaled the images to decrease dimension of data.
- **Augmentation:** Augmentation is used to increase variety of training data to decrease over-fitting, when augmentation is used, training data is repeated twice to increase number of samples.
- **Gender Information:** Gender information is used to check the effect of gender on bone age prediction. When gender information is used, it is fed to dense layer (100 nodes) then it is added to image information on flattening layer.
- **Convolution:** I created 6 stages of layers, first 5 stages includes convolutional layers (combination of VGG, Inception block and residual skip connections) to try different approaches to extract features. VGG blocks include 3*3 kernels stacked 1,2,3 times. If the 3*3 kernels stacked two consecutive times, it creates 5*5 kernel with less parameters and 1 additional activation function. If 3*3 kernels stacked 3 times, it creates a 7*7 kernel with 2 additional activation functions and less parameters.
  Inception blocks are the combination of 4 different paths to exploit different approaches to get details and sparse information. There are reduction layers before 3*3, 5*5 and pooling layers of inception block to reduce number of parameters and memory occupation. Residual connections help to learn identity functions and ensures that the next stage will learn at least as much as previous block through the flow of skip connection.
- **Pooling:** Max Pooling (2*2) is applied to reduce dimensionality of input and to preserve local features. In case of skip connections, the spatial size is reduced via strides of 2 by applying 1*1 kernels to skip path.
- **Flattening:** Flattening is a necessary step to convert high dimensional data into 1 dimension to feed into dense layers.
- **Dropout:** I inserted a dropout layer with a rate of 0.5 after the flattening to increase generalization of model and to reduce over-fitting.
- **Fully Connected layers:** After features extraction via convolution layers, the regression output is created through 3 dense layers (512,256,1 nodes) to exploit flattened input and decide the age of bones.
- **Batch normalization:** In order to reduce the effects of the vanishing gradient problem I optionally inserted batch normalization layers in f convolution layers to overcome covariance shift problem and to regularize network.
- **Output Layer:** The final layers with 1 node is basically a node without any activation function to use output as it is for regression problem.
- **Loss Function:** Mean Absolute Error is utilized to be consistent with previous studies of bone age prediction and MAE is more intuitive to human understanding.
- **Training:** Initial learning rate started as 0.1 then it dropped by factor of 0.1 when validation loss does not decrease for 4 consecutive epochs. The minimum learning rate is 0.001. In addition, early stopping of 8 epochs is used by monitoring validation loss, which means if validation loss does not increase in 2 consecutive reduction of learning rate, the further iterations are not necessary. The maximum epoch is set to 50. The batch size is 64 due to restrictions of memory. Adam optimizer was used: this Keras built-in method arranges dynamically the momentum permitting a faster and precise learning. The models were trained on NVIDIA GeForce GTX 1650 Ti GPU.

There are 5 different base models. All models start with 2 stages of convolutional layers

- stage 1: 3 convolutional layers with 3*3 kernels and 8 filters
- stage 2: 2 convolutional layers with 3*3 kernels and 16 filters
- 3 stages of layer of choice

model 1: basic convolution
model 2: inception block
model 3: skip connection
model 4: skip connection with pre-activation
model 5: inception block with skip connection

The number of filters increases consecutively as 32, 64 and 128. Then all filters are flattened and fed to the same dense layers. Then I created set of hyperparameters to check effect of gender information, batch normalization, drop-out and augmentation on these 5 models.

VI. Results

There are 5 base models with 4 different hyper parameter option. In total 25 models were trained and tested to check effects of gender information, batch normalization, data augmentation and drop-out regularization. The hyperparameter options are indicated Fig. 4.

Model #	Batch Normalization	Augmentation	Dropout	Gender
Model1	No / Yes	No / Yes	No / Yes	No / Yes
Model2	No / Yes	No / Yes	No / Yes	No / Yes
Model3	No / Yes	No / Yes	No / Yes	No / Yes
Model4	No / Yes	No / Yes	No / Yes	No / Yes
Model5	No / Yes	No / Yes	No / Yes	No / Yes

Fig. 4: Hyperparameters

Batch normalization helps overcome the covariate shift problem in deep learning models and enables better feature extraction, especially in Inception block-based models (model 2 and model 5). The use of batch normalization in these models led to a decrease in both training and validation mean absolute error (MAE), indicating its potential for use in deeper models. The results comparing models with and without batch normalization is given in Fig. 5.

The models that incorporate gender information and batch normalization showed the lowest training and validation mean

Model #	Batch Normalization	Train Loss	Val Loss	Test Loss
Model1	No	33.05	33.33	34.28
Model2	No	32.99	33.23	34.72
Model3	No	34.72	33.05	33.23
Model4	No	33.39	33.6	35.1
Model5	No	33.1	33.27	34.78
Model1_bn	Yes	33.05	33.18	33.77
Model2_bn	Yes	17.98	23.03	36.05
Model3_bn	Yes	33.3	33.5	34.9
Model4_bn	Yes	33.2	33.4	34.6
Model5_bn	Yes	25.8	35.2	37.6

Fig. 5: Models with/out Batch Normalization Results

absolute error (MAE) among all the models. Gender information plays a crucial role in the accuracy of bone age prediction.The results comparing models with and without batch normalization is given in Fig. 6.

Model #	Batch Normalization	Gender	Train Loss	Val Loss	Test Loss
Model1	No	No	33.05	33.33	34.28
Model2	No	No	32.99	33.23	34.72
Model3	No	No	34.72	33.05	33.23
Model4	No	No	33.39	33.6	35.1
Model5	No	No	33.1	33.27	34.78
nsferlearning ile gender kısmını da ekle öyle çalıştır.					
Model1_bn_gn	Yes	Yes	20.16	23	33.12
Model2_bn_gn	Yes	Yes	17.3	19.39	29.62
Model3_bn_gn	Yes	Yes	24.02	28.4	37.8
Model4_bn_gn	Yes	Yes	38.56	38.5	48.6
Model5_bn_gn	Yes	Yes	33.02	33.2	34.7

Fig. 6: Models with/out Batch Normalization and Gender

More specifically, models 1, 2, and 3 benefited from incorporating gender information. The combination of gender information and batch normalization improved the training and validation loss, but did not result in a decrease in the test loss. To address this, regularization techniques such as drop-out and data augmentation were used. This caused an increase in training and validation loss, but did not result in a decrease in the test loss.

Upon reviewing the validation and training graphs, I observed that the models ceased to learn after a brief number of epochs. The best validation epochs were between 3-50, with an average of 21. This suggests that the models are not complex enough to effectively differentiate between images and learn their features. Additionally, a review of the graphs for the 5 models (including batch normalization and gender information) revealed that, even though the average best epoch was 21, most of the learning process was completed after epoch 8. The learning process (number of epochs) vs loss graph is given in Fig. 7.

The Fig. 8 shows that models 2 and 5 (Inception block and Inception block with skip connection) have the lowest training losses. This could be due to the fact that they have more parameters than the other models as they contain an equal number of 3x3 kernels. However, the Inception blocks also include additional 1x1 and 5x5 kernels, as well as extra pooling, which increases the model complexity. While

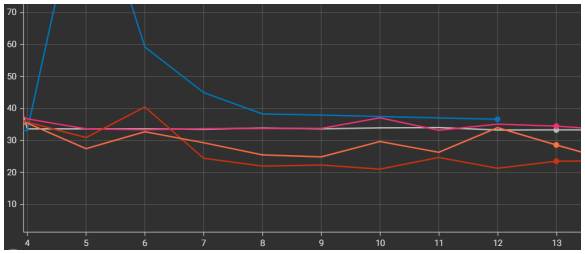

Fig. 7: Epochs vs Validation Loss

3x3 kernels focus on the finer details in the images, the 5x5 kernels help to understand more general features. This result motivated me to explore the relationship between MAE and model complexity by experimenting with more complex models.

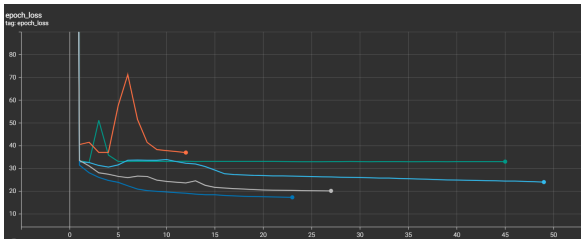

Fig. 8: Training Output vs Epoch

Since the models have high bias and low variance, meaning they are not complex enough, regularization techniques are not effective. To verify this, I applied transfer-learning using a ResNet-50 model and saw a decrease in the test MAE to almost 11 months. I also constructed a deeper model with more filters and multiple stages of Inception blocks, resulting in a further decrease in test MAE to 16 months.

Finally, I compared the models in terms of their training and prediction time. Models 2 and 5, which use Inception blocks, had longer training and prediction times compared to the other models. This is likely due to the fact that these models have 5.9 million parameters, while the other models have 3.5 million parameters. The additional parameters in the Inception blocks come from the extra kernels, as all filters are combined at the end of the Inception blocks. The Fig. 9 shows the comparison of training and prediction times in milliseconds per a batch.

In conclusion, while batch normalization helps address covariance shift and vanishing gradients, its impact is limited in the shallow models presented in this report. Adding gender information improved the models' ability to learn from the training images, but did not result in any changes to the test results. Due to the shallow nature of the models, regularization techniques such as dropout and augmentation did not reduce the test MAE. To fully assess the impact of regularization, the models should be made deeper.

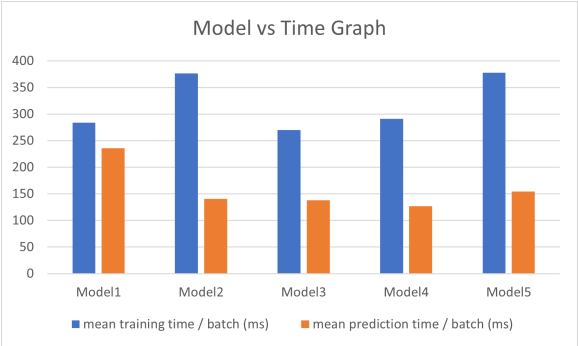

Fig. 9: Training and Prediction Time

VII. Concluding Remarks

In this study, I explored different types of convolutional neural network (CNN) models for estimating the age of children based on hand bone radiographs. I tested 5 different base models, including basic convolutional neural networks, VGG structure, inception blocks, original residual skip connections, pre-activation skip connections, and inception block with skip connections. These base models were trained with different hyperparameter choices such as using batch normalization, the addition of gender information, drop-out regularization, and data augmentation.

The models with a combination of inception blocks and batch normalization, along with gender information, showed the best results in terms of training and validation accuracy, and showed slight improvement in the test set. To improve the models, it is necessary to go deeper and to exploit the combination of inception blocks with skip connections. Additionally, gathering more information about the children from hospital records and using an ensemble approach to predict age could further decrease the mean absolute error between predicted age and the true age in months.

As I was working on this project as part of my university studies, I faced various challenges along the way. I initially implemented the project in Jupyter notebook, but as I started working with different models, things got messy, so I moved to using an integrated development environment (IDE) and a code versioning system. I encountered memory allocation errors many times and had to decrease the size of images to allocate them into memory.

I struggled with project planning and often pushed the memory allocation, causing many crashes during the training process. To overcome this, I created a metho to create more complex models by first training a shallow model to extract features from images, using it as a baseline transfer learning model, then freezing the first layers and training the last 2-3 layers again, also adding more convolutional layers. Through this method, I was able to decrease the test MAE from 30 to 16 months without having problem with memory allocation. However, I did not present these models in my final report as I did not have a proper experimental approach to compare them with a baseline.

Through this project, I gained valuable experience and knowledge. I challenged myself to create experiments with different models and read papers on image classification models to better understand their contributions to the field. I had the opportunity to develop deep learning models from scratch and test various architectures and hyperparameters. This allowed me to better understand the role and importance of the validation set during hyperparameter selection.

References

[1] D. B. Larson, M. C. Chen, M. P. Lungren, S. S. Halabi, N. V. Stence, and C. P. Langlotz, "Performance of a deep-learning neural network model in assessing skeletal maturity on pediatric hand radiographs," *Radiology*, vol. 287, no. 1, pp. 313–322, 2018.

[2] M. Elgendy, *Deep learning for vision systems*. Simon and Schuster, 2020.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, pp. 630–645, Springer, 2016.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.