Bone Age Prediction Using Convolutional Neural Networks and Visual Attention

Sebastiano Monti[†], Valentina Tonazzo[‡]

***Abstract*—This project revolves around the replication and enhancement of a neural network model originally employed in a successful bone age assessment contest using hand radiography and the exploitation of visual attention techniques. The overarching goal was to explore innovative architectural approaches that could simultaneously improve predictive performance and reduce computational complexity. One of the main aspects of this project was the exploration of three prominent pre-trained architectures: InceptionV3, Xception, and a customized variant of ResNet50 featuring overlapping patches. Beyond the exploration of pre-trained architectures, a pivotal contribution of this work lies in the implementation of a Convolutional Neural Network from scratch, composed of three distinct branches. These branches are designed to combine global feature extraction with local feature extraction using an attention mechanism. By integrating this novel approach, the model aims to capture both the generic context and the fine-grained details, essential for accurate bone age assessment. The comparative analysis of these architectures during the testing phase yielded invaluable insights into their strengths and weaknesses. Additionally, an exploration into the role of patch dimensions revealed a compelling finding: larger overlapping patches may play a crucial role in enhancing the model's overall understanding of the image. This revelation underscores the importance of patch size in medical image analysis and its potential implications for future research. To further leverage the strengths of each architecture, an ensemble approach was employed achieving more robust predictions and less overfitting.**

***Index Terms*—Convolutional Neural Networks, Bone Age Prediction, Computer Vision, Visual Attention, Image Classification.**

I. Introduction

During the growth of a child, bones change in size and shape and a deviation of the bone age from the chronological age may address the presence of a growth problem. Bone age assessment is a common clinical practice that estimates the maturity of skeletal system and is used in order to diagnose endocrine disorders in children and adolescents. The adopted standard for bone age assessment is based on two most relevant techniques. The Greulich-Pyle method [1] and the Tanner Whitehouse method [2]. Being these human supervised techniques, they require a significant amount of time and workload from experienced medical personnel to be conducted. Also, they can face the influence of doctors with different standards. The problem of bone age assessment using automated approaches received therefore increasing attention in the last years, in that it could for sure minimize judgment time, but also standardize the evaluation removing the human variability component. Of course still requiring doctoral supervision and expertise.

Many studies have shown that deep learning techniques can be successfully used in the domain of medical imaging [3], [4]. In particular, in the field of bone age assessment, studies have shown that convolutional neural network (CNN) based techniques can achieve predictions whose accuracy is similar to that of an expert radiologist [5].

The purpose of this report is to show the different approaches that have been followed in facing the problem of bone age prediction and compare the implemented models in order to evaluate if one is able to outperform the others. The adopted techniques have in common the usage of CNNs, differing however in the processing of the image dataset. One of the implemented approaches regarded the age prediction over the radiographs in their entirety, while the other consisted in processing multiple patches of the original images. Moreover, the sex information about the candidates was added to the radiographs, hoping that this would improve the accuracy of the predictions.

II. Related Work

In the domain of pediatric radiology, accurate assessment of skeletal maturity, often determined by bone age estimation, plays a pivotal role in clinical practice. Traditionally, this task has relied on manual assessment by radiologists, which is subject to inter-observer variability and time-consuming procedures. The emergence of deep learning techniques has introduced the potential for automating and enhancing the precision of skeletal maturity assessments through the analysis of pediatric hand radiographs. The *"Performance of a Deep-Learning Neural Network Model in Assessing Skeletal Maturity on Pediatric Hand Radiographs"* [5] presents a study aimed to compare the performance of a deep-learning-based bone age assessment model using hand radiographs with that of expert radiologists and existing automated models. A total of 14,036 clinical hand radiographs and corresponding reports were obtained from two children's hospitals for model training and validation. In the first test set, comprising 200 examinations, the model's bone age estimates were compared with the mean estimates from clinical reports and three human reviewers. The assessment of the model's performance included comparing the root mean square (RMS) and mean absolute difference (MAD) between the model's estimates and the reference standard bone ages.

[†] Department of Physics and Astronomy, University of Padova, email: sebastiano.monti@studenti.unipd.it

[‡] Department of Physics and Astronomy, University of Padova, email: valentina.tonazzo.1@studenti.unipd.it

The *"RSNA Pediatric Bone Age Machine Learning Challenge"* [6] aimed to showcase the role of artificial intelligence (AI) in medical imaging. Participants were tasked with creating machine learning (ML) algorithms to determine pediatric skeletal age from hand radiographs. The challenge attracted 260 participants, with the top-performing models achieving a mean absolute distance of 4.2 to 4.5 months. This collaborative approach demonstrated the potential of ML in improving diagnostic accuracy and patient care. In particular the winning approach utilized both pixel and gender information within a single neural network, using 500×500-pixel images. It employed the Inception V3 architecture for pixel data and concatenated it with gender information. In this paper is shown a computational implementation of the latter model, reaching reasonable results, and setting it as a baseline model for further comparison.

This work mainly differ from the previous ones since the neural network was then modified in order to make it computationally less expensive but at the same time more performative: in order to do this a neural network called Xception was explored, first presented in the *"Xception: Deep Learning with Depthwise Separable Convolutions"* [7].
The second-place approach involved training sex-specific models using contrast-enhanced image patches of size 224x224 pixels, which were subdivided into 49 overlapping patches. Transfer learning and fine-tuning of ResNet-50 architectures pretrained on ImageNet were used in this approach. Also in this case the role of patches was further explored and analyzed to achieve a reacher comprehension of the patch utility. Finally a tecnique of ensamble voting was used to enhance the overall performances and draw a comparison between all the implemented methods.

The paper *"Diagnose like a Radiologist: Attention Guided Convolutional Neural Network for Thorax Diseas classification"* [8] introduces a novel approach to thorax disease classification on chest X-ray images, addressing limitations of existing methods. Traditional methods utilize the entire image for network learning, which is problematic due to irrelevant noisy areas and irregular borders hindering network performance. To overcome these issues, the paper proposes a three-branch attention guided convolution neural network (AG-CNN). The AG-CNN learns from disease-specific regions to avoid noise and improve alignment, integrates a global branch to compensate for lost discriminative cues by the local branch, and conducts comprehensive experiments on the ChestX-ray14 dataset. Results indicate that AG-CNN improves the average AUC from 0.841 to 0.868 with ResNet50 and achieves a new state of the art average AUC of 0.871 with DenseNet-121. This approach may be useful also for the task of bone age prediction due to the similarity of its purpose, although applied to a regression task rather than a classification one.

III. Processing Pipeline

A first step of this project consists in downloading the image RSNA train, validation and test sets and the associated metadata from the official website, which includes pediatric hand radiographs, gender information, and bone age labels. The dimension of the train set were over 10 GB of memory, so it was necessary to reduce images' dimensions and make the dataset more tractable. Firstly, each radiographic image was resized to a consistent size of 500x500 pixels. Simultaneously, the metadata have been processed. It is essential to appropriately format this data for model's training and evaluation.

Data generators play a pivotal role in organizing and batching the images, so they were used to prevent memory overusage. In traditional TensorFlow workflows, data generators are commonly associated with image classification tasks, where they efficiently load and preprocess image data for model's training. However, in this project, the focus shifted to a regression task, which necessitated a departure from the typical image-centric data generators. Given the nature of regression tasks, it became imperative to introduce custom data generators that not only handle data loading but also incorporate techniques such as data augmentation. Additionally, a specialized data generator was designed to extract patches from each image, which was useful for patch-based ResNet50 model. Pre-trained models are trained on large-scale datasets (ImageNet, in our case) that consist of three-channel images (typically RGB), as a result, when utilizing this type of models in a new task or dataset, it's essential to adhere to the expected input data format in order to align with the model's architecture and ensure compatibility with the learned representations. In contrast, when working with the non-pretrained model, there was no inherent reason or requirement to utilize three-channel images. So, to streamline the data preprocessing pipeline and optimize the model's training process, a dedicated data generator was implemented to handle single-channel images. Inside the dataloader, images are augmented to enrich the dataset's diversity.

This project involves the implementation and training of three distinct neural network models, based on different pre-trained deep neural networks, used as a baseline comparison with a three branch CNN featuring attention mechanism, implemented and trained from scratch.

During the training processes of the three pre-trained models, *ReduceOnPlatou* and *Earlystopping* callbacks were used to improve training efficiency and prevent overfitting. A learning rate scheduler was also implemented, considering an exponential decay from a value of 10^{-3} to 10^{-5} within each epoch, without however displaying a real improvement over the losses and resulting instead in slower performance improvements. For this reason the learning rate was initially fixed to 10^{-3} and eventually automatically reduced with the *ReduceOnPlatou* callback in case the validation loss stopped to improve. On the other hand, we implemented a customized training loop for the AG-CNN architecture. In this implementation, we set a fixed learning rate of 10^{-5}. This decision was based on two primary reasons: firstly, during our initial trials with a learning rate set to 10^{-3}, we observed poor performance in terms of both training and validation loss. The losses at the first epoch started around 10^{6} and just

Set Type	Number of Samples	Female %	Average Age
Training	12611	0.46	127
Validation	1425	0.46	127
Test	200	0.5	132

TABLE 1: Summary of dataset subdivision.

decreased down to 10^1 after 7 epochs, indicating inadequate convergence and suboptimal learning. Secondly, referring to Q. Guan et al. [8], the authors used a learning rate of 10^{-8}. However, adopting such an extremely small learning rate led to excessively time-consuming training trials. Thus, we opted for a compromise by setting the learning rate at 10^{-5} to strike a balance between convergence speed and computational efficiency.

Each model was trained for 7 epochs, while the AG-CNN was trained for 16 epochs. The validation set serves as a critical reference to monitor training progress. In the compiling step, Adam was chosen as an optimizer, Mean Squared Error (MSE) as loss and Mean Absolute Error (MAE) in months as accuracy. Each model was then evaluated using the designated test set. Essential evaluation metrics were computed, including MAE, root mean square error (RMSE), and correlation coefficients, to gauge models' accuracy in age predictions.

IV. Signals and Features

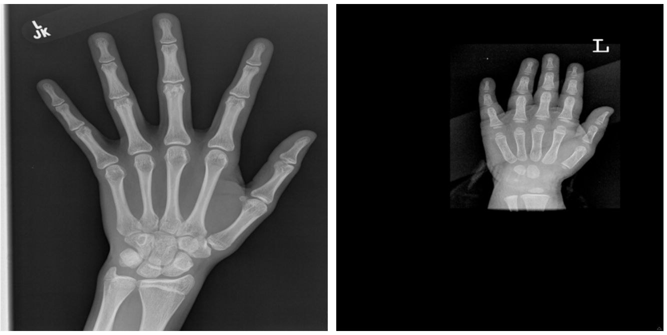
The dataset that have been considered for this project is the *2017 RSNA Pediatric Bone Age Challenge Dataset*. It contains a total of 14236 `.png` pediatric hand radiographs subdivided as shown in Table 1.

Each dataset is accompanied by a `.csv` file containing for each image, the sex of the candidate and his or her ground truth bone age expressed in months, spanning from a minimum of 1 months to a maximum of 228 months.

The same preprocessing pipeline was applied to all the subsets and involved the following steps. Firstly, in order to have images with same dimensions, but also to reduce computation times and memory expense, all images have been resized to a dimension of 500×500, following the same choice of RSNA challenge's winners [6]. Two examples from the resized training set can be seen in Figure 1.

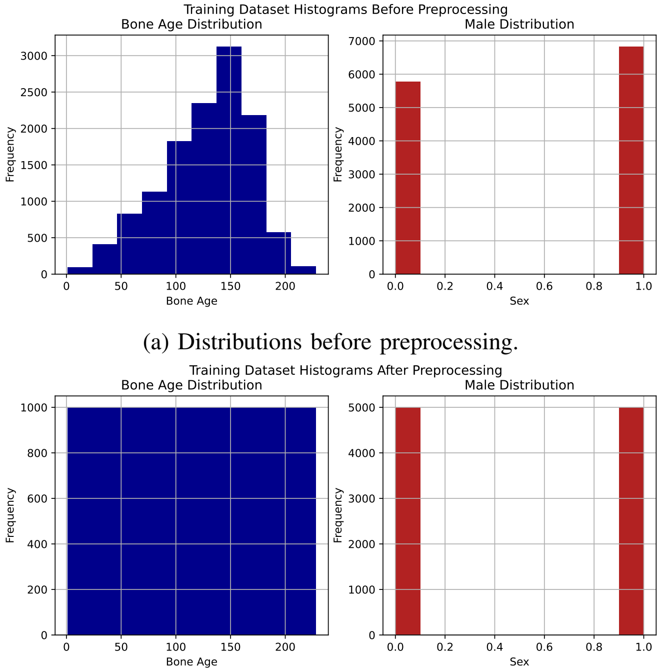
By examining the distribution of the data in terms of bone age and sex of the candidates, it is possible to observe a similar trend between all the datasets. Histograms of these distributions for the training dataset are shown in Figure 2a.

Preliminary trials showed that, as one may expect, the model learns to predict with higher accuracy the bone age from samples that are present with a higher frequency, with respect to the ones that instead lays on the tails of the histogram. For this reason, it was decided to randomly re-sample the images from each dataset (allowing repetitions), in order to obtain a uniform distribution both in the bone age and in the sex of the candidates. The training set histograms after re-sampling are depicted in Figure 2b. This allowed to obtain a significant improvement of the loss in each of the implemented architectures.



(a) Bone age = 180 (b) Bone age = 33

Fig. 1: Examples of images after resizing. Dimensions are 500×500.



(a) Distributions before preprocessing.

(b) Distributions after preprocessing.

Fig. 2: Training dataset's bone age and sex distributions.

Having this done, a normalized version of the bone age was calculated by subtracting to each age value, the average over the re-distributed dataset and dividing by its standard deviation. Then, a dataloader was created by following two different approaches.

- **Image data augmentation:** the dataloader generates randomly shuffled batches of 8 images, respectively associating to each of them the information about sex of the candidate and the bone age as label. Images are converted from a single channel to RGB, then transformed from a representation in interval $[0, 255]$ to one in $[-1, 1]$ and finally augmented using the augmentation parameters summarized in table 2.

- **Patches data augmentation:** images are converted to RGB and randomly shuffled, then 3 patches of dimensions 224×224 are randomly extracted from each of them. Each patch is associated to the respective sex of the candidate and to the bone age label. Data augmentation is also applied, although in this case just a rotation range of 5° and horizontal flip were considered. In this case, generated batches contained a total amount of 24 images, corresponding to 8 different candidates (i.e. 3 patches × 8 candidates).

Augmentation Parameter	**Range**
Rotation	5°
Width shift	0.1
Height shift	0.1
Shear	0.01
Zoom	0.25
Horizontal flip	True

TABLE 2: Image data augmentation parameters.

V. Learning Framework

A. *Inception-V3 model*

The proposed model (Figure 3a) is based on the RSNA bone age challenge's winning architecture. The input is formed by two different layers:

- Gender Input: This layer receives the gender information as a single value (0 for male, 1 for female) and is designed to accommodate batch processing.
- Image Input: accepts RGB images of size 500x500 pixels, representing pediatric hand radiographs, processed in batches for efficiency.

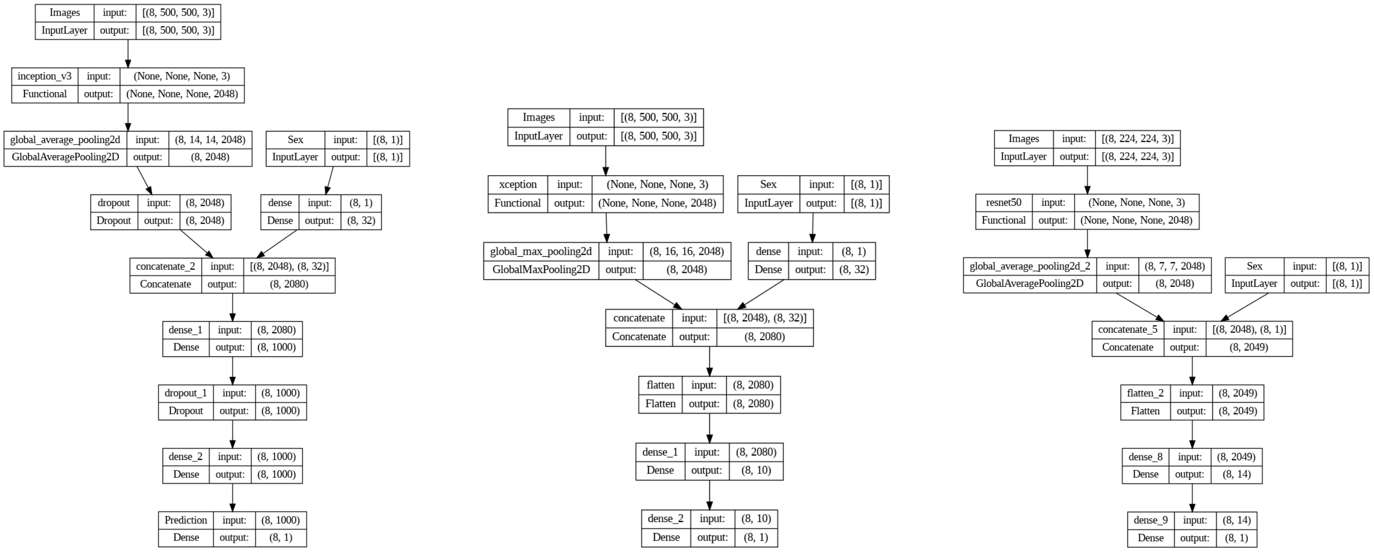
A dense layer with 32 units and a sigmoid activation function is applied to the gender input to capture and encode the sex-related information. The model utilizes the InceptionV3 architecture, pretrained on the ImageNet dataset, as a feature extractor. Transfer learning is employed, with the layers of the InceptionV3 frozen to retain the pre-trained features. The global average pooling layer aggregates the feature maps generated by the InceptionV3. Finally, two densely connected layers with 1000 units and a hyperbolic tangent (tanh) activation function further processes the joint features. The final prediction layer consists of a single unit with a linear activation function, producing the estimated skeletal age as a continuous numeric value. Between the last trainable layers dropout layers were added in order to prevent overfitting.

B. *Xception model*

In a similar way to the one described for the previous network, the first input layer of the Xception-based model (Figure 3b) is dedicated to gender-related data, accepting a single numerical value representing the individual's male or female variable. This input is processed through a dense layer with sigmoid activation, allowing the network to learn gender-specific features. The image input layer, again, accepts images with a resolution of 500x500 pixels and 3 color channels (RGB), since these images serve as the primary source of visual information for age estimation. Notably, in contrast with the previous network, now pre-trained Xception has been used; initialized with ImageNet weights to extract features from the images. Xception, short for ”Extreme Inception”, [7] is a neural network architecture that was introduced as an evolution of the Inception architecture. It was designed to achieve state-of-the-art performance in computer vision tasks, particularly in image classification and object recognition. Xception draws inspiration from the Inception architecture, particularly the idea of using multiple parallel convolutional filters with different receptive field sizes. However, instead of traditional convolutions, it primarily utilizes depthwise separable convolutions, which are more efficient . This means it separates the spatial convolution (depthwise convolution) and the pointwise convolution (1x1 convolution) into two separate operations. This helps reduce the number of parameters and computation, making the network faster and more efficient. Fine-tuning of the Xception model is disabled by setting its layers to non-trainable. As in the previous network, Xception typically uses global average pooling to reduce spatial dimensions. The extracted image features and the processed gender-related information are concatenated to form a joint feature representation. Then, the joint feature representation flattened and passed to a dense layer with 10 units and ReLU activation. These intermediate layers help the network capture complex relationships between the input data. As a main difference with respect to InceptionV3-based model, here there are no droupout Layers, reducing the complexity of the neural network. The final prediction is obtained through a dense layer with a single output unit and linear activation. This output represents the estimated bone age of the subject. The whole network is formed by a total of 20.882.365 (79.66 MB) parameters, which is quite similar to the amount of the Inception-V3 model, but just 20.885 (81.58 KB) of them are trainable, making it less computationally expensive.

C. *ResNet50 model*

In a similar way as the above described models, here we explore the ResNet50-based architecture (Figure 3c). ResNet50 is a deep neural network with 50 layers. These layers are organized into blocks, with each block consisting of multiple convolutional layers. The central innovation of ResNet50 is the usage of residual blocks, which include shortcut connections (skip connections or identity shortcuts) that skip one or more layers [9]. Whitin a residual block, the output of a layer is added to the output of a previous layer, effectively creating an identity mapping which ensures that the network can always learn the identity function as a minimum requirement. Any learned transformation is applied as an adjustment to this identity. In this way the architecture helps address the vanishing gradient problem, making it easier to train very deep networks. The main difference with the previous presented models is in the input of the network. In fact, input images in this case are 3 (224×224) overlapping patches of the original


Images | input: | [(8, 500, 500, 3)]
InputLayer | output: | [(8, 500, 500, 3)]
inception_v3 | input: | (None, None, None, 3)
Functional | output: | (None, None, None, 2048)
global_average_pooling2d | input: | (8, 14, 14, 2048)
GlobalAveragePooling2D | output: | (8, 2048)
Sex | input: | [(8, 1)]
InputLayer | output: | [(8, 1)]
dropout | input: | (8, 2048)
Dropout | output: | (8, 2048)
dense | input: | (8, 1)
Dense | output: | (8, 32)
concatenate_2 | input: | [(8, 2048), (8, 32)]
Concatenate | output: | (8, 2080)
dense_1 | input: | (8, 2080)
Dense | output: | (8, 1000)
dropout_1 | input: | (8, 1000)
Dropout | output: | (8, 1000)
dense_2 | input: | (8, 1000)
Dense | output: | (8, 1000)
Prediction | input: | (8, 1000)
Dense | output: | (8, 1)
Images | input: | [(8, 500, 500, 3)]
InputLayer | output: | [(8, 500, 500, 3)]
xception | input: | (None, None, None, 3)
Functional | output: | (None, None, None, 2048)
Sex | input: | [(8, 1)]
InputLayer | output: | [(8, 1)]
global_max_pooling2d | input: | (8, 16, 16, 2048)
GlobalMaxPooling2D | output: | (8, 2048)
dense | input: | (8, 1)
Dense | output: | (8, 32)
concatenate | input: | [(8, 2048), (8, 32)]
Concatenate | output: | (8, 2080)
flatten | input: | (8, 2080)
Flatten | output: | (8, 2080)
dense_1 | input: | (8, 2080)
Dense | output: | (8, 10)
dense_2 | input: | (8, 10)
Dense | output: | (8, 1)
Images | input: | [(8, 224, 224, 3)]
InputLayer | output: | [(8, 224, 224, 3)]
resnet50 | input: | (None, None, None, 3)
Functional | output: | (None, None, None, 2048)
global_average_pooling2d_2 | input: | (8, 7, 7, 2048)
GlobalAveragePooling2D | output: | (8, 2048)
Sex | input: | [(8, 1)]
InputLayer | output: | [(8, 1)]
concatenate_5 | input: | [(8, 2048), (8, 1)]
Concatenate | output: | (8, 2049)
flatten_2 | input: | (8, 2049)
Flatten | output: | (8, 2049)
dense_8 | input: | (8, 2049)
Dense | output: | (8, 14)
dense_9 | input: | (8, 14)
Dense | output: | (8, 1)

(a) InceptionV3-based architecture (b) Xception-based architecture (c) Resnet50-based architecture

Fig. 3: Detailed description of the implemented architectures.

augmented images. By analyzing small, localized regions, the network can extract local features and patterns, which can be crucial for recognizing complex structures. The output from ResNet50 is passed through a Global Average Pooling 2D layer, which helps reducing the spatial dimensions, while the gender information input and the processed image features are concatenated together. In this case the gender information is not previously processed before the concatenation. The concatenated features are then flattened, followed by a dense layer with 14 units and a ReLU activation function. Finally, a single dense layer with a linear activation function produces the output prediction.

D. AG-CNN: attention guided model

The AG-CNN architecture we implemented in this project is inspired on the one developed by Qingji Guan et al. for thorax disease classification [8] and is composed by three major branches: Global, Local and Fusion Branches. A visual description of the model is represented in Figure 4.

All the designed networks perform a regression task, giving as final output the predicted bone age associated to each image. The architecture of the first two branches is the same and is based on a ResNet50 backbone, implemented from scratch. Our implementation of ResNet50 consists of five convolutional blocks. The first one is composed by a convolutional layer of 64 7×7 filters, followed by batch normalization, LeakyReLU activation function and a max pooling layer, while the others encapsulate a series of convolutional layers with progressively increasing number of filters organized in a ”bottleneck” design, stacking together 1×1, 3×3 and 1×1 convolutions. This is done for practical reasons, since bottleneck ResNets are more computationally economic compared to non-bottleneck ones [9]. Convolutional layers' weights are all initialized using He uniform initialization, which is proven to provide good performances when combined with rectified linear activation units in visual recognition tasks [10]. Batch normalization is then applied after each convolutional layer in order to standardize the outputs and stabilize the learning process, followed by LeakyReLU activation functions. The last convolutional block is then followed by a global average pooling, outputting an array of 2048 features for each input image, which is kept in memory in both Global and Local Branches to be subsequently fed as input to the Fusion Branch. The pooling output is finally passed to a 256 units dense layer with He uniform weight initialization and LeakyReLU activation function, after which we applied a 0.5 rate dropout and finally, to a dense layer with 1 output unit and linear activation, predicting the bone age. What differentiates the Global from the Local Branch is that, while the inputs of the first branch are the original single-channel images, the second is instead fed with attention images: local crops of the original images, extracted by processing the output features from the last convolutional block of the global branch (more on this in subsection V-D1). The Fusion Branch instead, as partially stated above, takes in input the concatenation of Global and Local Branches' average pooling outputs (two arrays of sizes

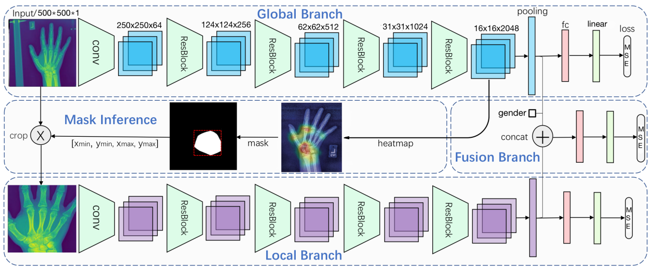


Fig. 4: Scheme of the AG-CNN architecture (Image taken from [8] and re-adapted).

$batch\ size \times 2048$) and the gender information (an array of size $batch\ size \times 1$). The resulting array passes through two dense layers having respectively 256 and 128 output neurons, He uniform weight initialization and LeakyReLU activation function, both characterized by a 0.5 rate dropout. The output is finally computed by a dense layer with 1 output unit and linear activation function, providing the bone age prediction.

1) Data pipeline: Pre-processed single channel (500×500) images are fed to the Global Branch. After the last convolutional block, the extracted features follow two different paths: on one side, they are passed to the remaining layers of the branch until the end, in order to get the Global Branch's predictions of the bone age from each image; on the other side, the same feature maps are also fed to a function which aims to process them to infer attention regions from original radiographs. In particular, the processes the feature maps go through, first involve a reshaping from $(16 \times 16 \times 2048)$ to (256×2048) tensors, that are then summed along channel axis and reshaped back into a (16×16) matrix. The entries of this matrix are then rescaled between 0 and 255 and the matrix is up-sampled to the size of the input radiographs. A heatmap is then computed by applying Otsu's binary thresholding. Features with values above the automatically determined threshold are set to 255 (white), indicating high importance, while ones below the threshold are set to 0 (black), indicating low importance. This results in several connected components on each heatmap. At this point, the connected components undergo through a labeling process and then the connected component having the widest area in the heatmap is identified based on the sum of its pixel values. If no connected components are found, the attention mask is set to an empty array. The actual attention mask, which represent the core of attention mechanism, is obtained by multiplying the binary heatmap with the selected connected component and the bounding box coordinates of the attention mask are determined in a way that the mask is entirely contained inside the box. Finally, the corresponding region of interest (ROI) is extracted by cropping the original image around the bounding box's coordinates. The ROI is up-sampled to the size of the input radiographs and normalized between -1 and 1 in order to be fed to the Local Branch, which will output its own prediction of the bone age in the same way as the Global branch did, but supposedly this time, ignoring all unimportant information contained in the original images. It is important to highlight that since ROIs depend on feature maps directly outputted from the Global Branch, they are not pre-determined images, but instead are learned during the training process. Thus, the network aims at understanding which are the most meaningful spacial areas that progressively improve the performances of the whole model. This represent the core application of attention mechanism in the visual task we aim to accomplish. Finally, the last pooling layers' outputs, coming from both Global and Local Branches, are concatenated together with the gender information and fed to the Fusion Branch, which produces its own prediction of the bone age.

2) Training and validation strategy: As a main difference with the previously implemented models, in this case a customized training loop has been designed in order to manually control the flow of images and the the updates of each branch. Hence, instead of using the usual Keras' `model.fit()` attribute, a `train_step()` and a `validation_step()` functions for each branch have been defined.

In the first one, the model's parameters have been set to trainable, which means a tape will automatically track any operations involving these variables for the purpose of calculating gradients. Each Branch is called inside a tape, returning its own output prediction of the bone age. Then, three loss functions, one for each Branch, are calculated with respect to the input's ground truth labels. Similarly to the previous cases, the chosen loss function is the Mean Squared Error.

$$\mathcal{L} = MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i^{pred} - y_i^{true})^2$$

Additionally, another performance metric, the Mean Absolute Error in months, is computed between the predicted bone ages and the actual ground truth labels.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\bar{y}_i^{true} - \bar{y}_i^{pred}|$$

In this expression, differently from MSE, the bone ages are not manipulated in their normalized version y, but instead transformed back in months units, using the average μ_{age} and the standard deviation σ_{age} computed in the phase of bone age normalization, during the pre-processing of the data:

$$\bar{y} = \mu_{age} + \sigma_{age}y$$

At this point, following what is done in [8], the overall AG-CNN's loss is computed as a convex combination of the losses of the three Branches in order to assign a different importance to each of them, in the following way:

$$\mathcal{L} = 0.8\mathcal{L}_{Global} + 0.1\mathcal{L}_{Local} + 0.1\mathcal{L}_{Fusion}$$

Three gradients are then computed, one for each component of the overall loss, with respect to the trainable parameters of the associated Branch, weighted by the respective factor (this approach is different to the one followed by Guan et al., in that they computed the gradients of the overall loss with respect to the ensemble of all trainable variables of the network). Then, three Adam optimizers (one for each Branch) with constant learning rate set to 10^{-5} apply computed gradients to update the trainable parameters of each respective Branch. It is important to highlight that the three Branches do not share weights, since they have distinct purposes.

On the other hand, the `validation_step()` follows the same logic but, in this case, the model's parameters are not set to trainable, hence they are not updated. The performances of the model at each epoch are evaluated on the basis of the validation loss of the Fusion Branch $\mathcal{L}_{Fusion}^{Validation}$ with respect to the one of the previous epochs. The Fusion Branch should in fact put together all the salient features collected by the other two Branches processing the images, plus the gender information. Each training epoch consists in applying the implemented `train_step()` to each batch of the training dataset, following the pipeline explained in section V-D1. At the end of the epoch, the same is done applying the `validation_step()`. As already said, the difference with the previous implemented models is that the overall AG-CNN architecture have been trained from scratch without the usage of pre-trained neural networks.

Model Type	Total parameters	trainable parameters	Non-trainable parameters
InceptionV3	24.885.849	3.083.065	21.802.784
Xception	20.882.365	20.885	20.861.480
ResNet50	23.616.427	28.715	23.587.712
Global Branch	22.984.449	22.934.401	50.048
Local Branch	22.984.449	22.934.401	50.048
Fusion Branch	1.082.113	1.082.113	0

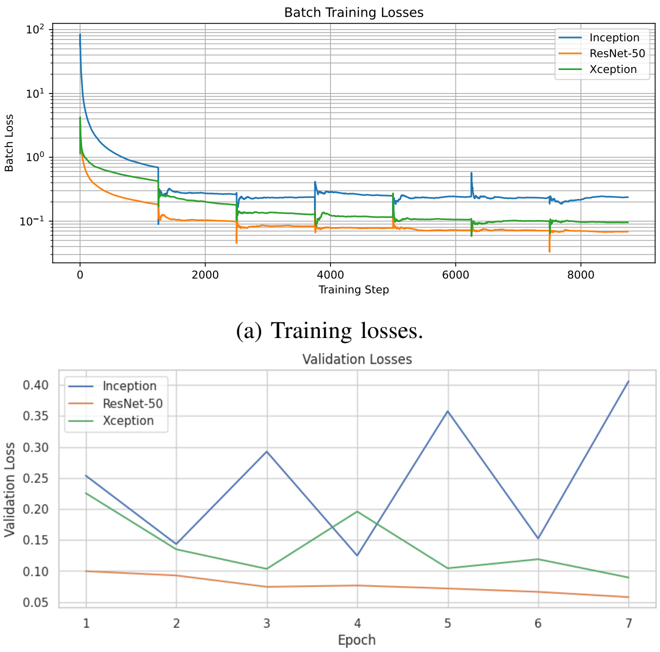
TABLE 3: Comparison of different architecture's parameters.

VI. Results

In this section, the obtained results over the training of the implemented models are summarized. The main tested conditions regard the variation of the dataloader (i.e. analysis of complete augmented images or analysis of augmented patches of the original radiographs) or the variation of the core models (Inception-V3, Xception, ResNet50 or AG-CNN). In Table 4 the main training configurations are summarized.

A. Analysis of pre-trained models

The batch and validation losses during the training process for each model are respectively shown in Figures 5a, 5b. From the plots it can be observed that the validation loss exhibits an oscillatory behaviour, apart probably for the ResNet50-based model, where performances would may have benefited from a longer training. In general, the training and validation losses observed across three distinct pre-trained models demonstrate a striking similarity in their behavior. Despite their different architectures and pre-trained weights, they exhibit a consistent pattern. This behavioural consistency underscores the robustness of the pre-trained models and suggests a common underlying learning dynamics, despite their initial differences.



(a) Training losses.

(b) Validation losses.

Fig. 5: Loss plots of pre-trained models.

B. Analysis of radiographs' patches

Since analyzing patches gives more freedom in the adjustment of patch size and number of extracted patches from each radiography, a fine tuning test was done in order to assess which combination of these two parameters led to the best performances on the training loss over one training epoch. This in-depth analysis was conducted on ResNet50-based model for two main reasons:

- It is the model that displays the best performances among the analyzed pre-trained architectures;
- The authors of the second winning place in [6] trained a ResNet50 architecture on overlapping patches.

In Table 5, the considered combinations are summarized, while the results of the training losses are shown in Figure 6. What is possible to observe from this graph is that when numerous small patches are generated from the images, the model's ability to generalize (i.e., perform well on unseen data) decreases. This implies that using a large number of small patches might not yield optimal results in terms of model performance. Conversely, when a smaller number of larger patches are used, the model tends to generalize better. However, there is not a significant difference in terms of loss between 224 × 224 and 500 × 500 pixels patch sizes. This suggests that within this range, the model's performances remain relatively similar. As a consequence, opting for the

Data Size	Core model
500 × 500 Images	Inception-V3
500 × 500 Images	Xception
224 × 224 Patches	ResNet50
500 × 500 Images	AG-CNN

TABLE 4: Summary of considered training configurations.

Number of patches	Patch size
1	500 × 500
3	224 × 224
9	128 × 128
36	64 × 64
150	32 × 32

TABLE 5: Tested fine tuning parameters for patches' analysis.

smaller size can offer advantages in terms of computational resources and training speed. This also justifies the initial choice of setting the patch number and size, respectively to 3 and 224×224.

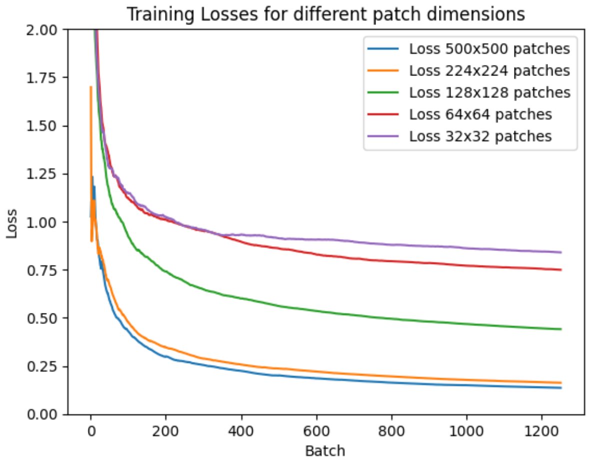
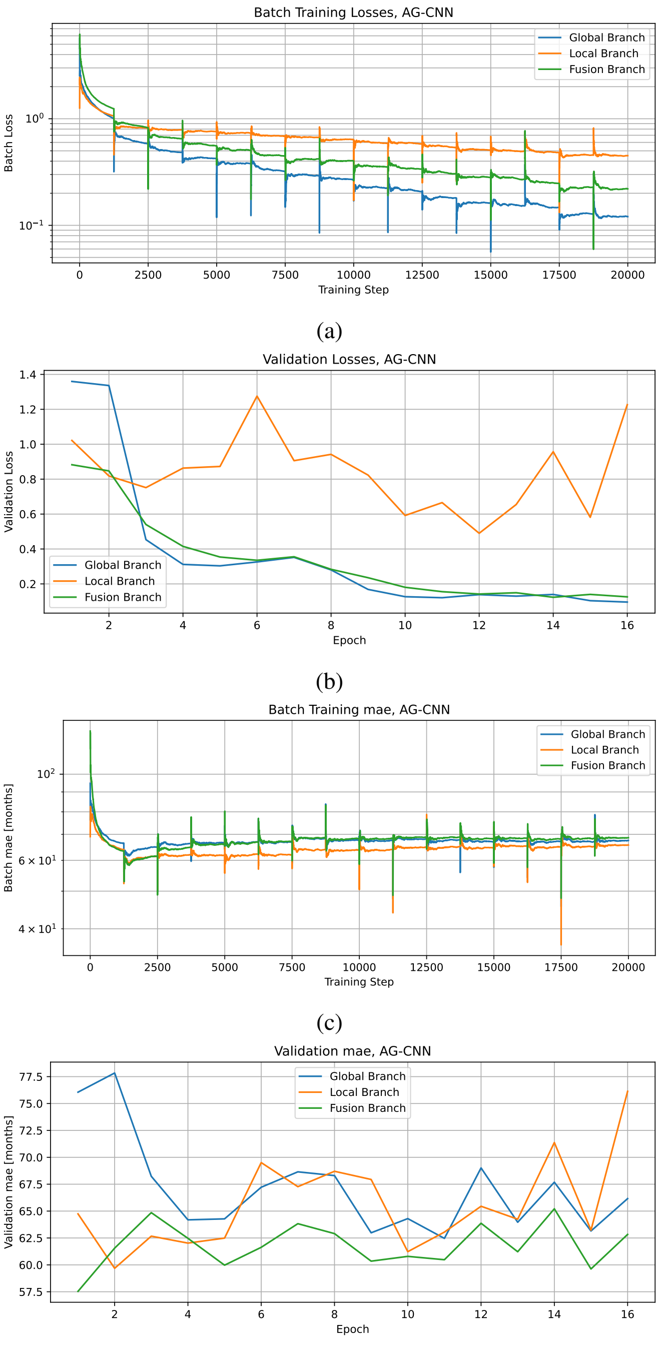


Fig. 6: Plot of the losses over one training epoch for the fine tuning over patch number and size.

C. *Analysis of AG-CNN*

The central focus of this project revolved around the implementation and training of an Attention-Guided Convolutional Neural Network from scratch. During this process, in a similar way of the previous discussion, we analyzed the behavior of various metrics such as training loss, validation loss, training MAE, and validation MAE across the three Branches that compose the whole network architecture. The metrics with the higher relevance are the ones of the Fusion Branch, since it should put together and make a prediction based on the information contained in the original and the attention images, elaborated by the other two Branches, plus the gender information.

Upon plotting these metrics after 16 epochs of training, a distinct pattern emerged. At the beginning of the training process, the Fusion Branch's batch loss started at higher values compared to the losses of the two other Branches, as can be seen in Figure 7a. However, it gradually decreased below the loss of the Local Branch already after the second epoch. The batch losses of the three Branches seem to keep decreasing also during the last epochs, indicating that the network's performances may have benefited from a longer training. Furthermore, the observed asymptotic behavior of the three batch loss curves raised the possibility of a bias error. This phenomenon, implies that the model may be exhibit a systematic deviation from the ground truth due to inherent biases in the data or model architecture. The same behavior is also observed in the training MAE, in Figure 7c. Among the three Branches, a striking similarity in their trajectories emerges. However, it becomes evident that the Local Branch consistently achieves the overall best performance.



(a)

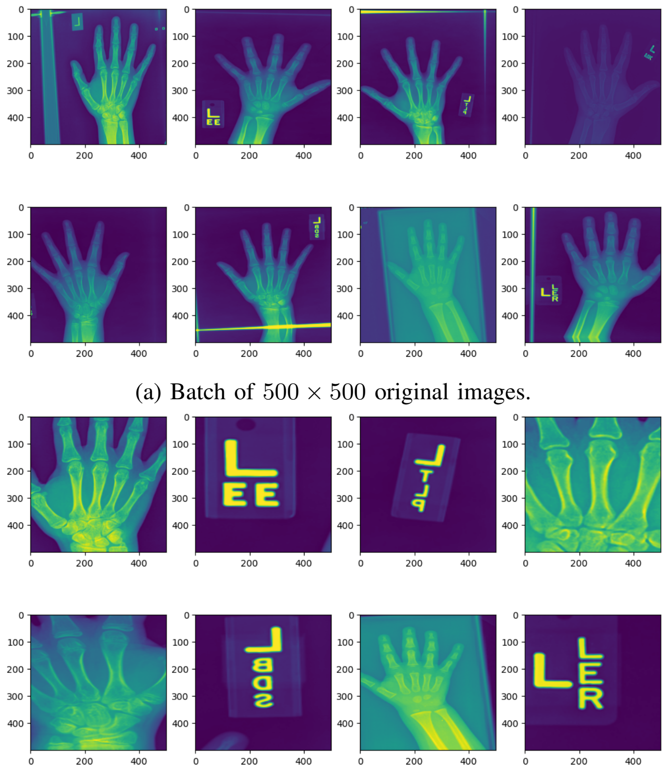
(b)

(c)

(d)

Fig. 7: AG-CNN training and validation metrics.

On the other hand, during the validation phase of our analysis, displayed in Figures 7b and 7d, the validation loss of the Fusion Branch proved to behave very closely to the one of the Global Branch. The Local Branch instead, exhibited the least favorable performances in terms of both validation loss and MAE, displaying worst results and a kind of erratic trend. These behaviours may be due to the fact that the Fusion Branch learns to give less importance to the predictions of the Local Branch, concentrating mainly on the input provided by the Global Branch; but also, that probably something is not



(a) Batch of 500 × 500 original images.

(b) Batch of 500 × 500 attention images.

Fig. 8: batch of images after 16 epochs of AG-CNN training.

working as expected during the training of the Local Branch. In fact, by examining a batch of attention images at the last training epoch and comparing it with the original batch of images, as displayed in Figure 8, it is clear that in some cases, when present, the model drives its attention to the letters in the background of the original images. Driving thus the attention to a completely non informative content, thus hindering the Local Branch from improving.

Notably, the Fusion Branch managed to achieve better results during validation. In fact, in some cases, it gets slightly lower Global Branch's validation loss, indicating its ability to generalize and perform competitively in certain scenarios.

D. Comparison between the models

To better compare the implemented architectures, their performances have been assessed over their output predictions on the test set and then compared to the real bone age values. Furthermore, in order to derive the average behaviour of the implemented pre-trained models, an ensemble prediction was computed by performing a weighted average over the corresponding bone age predictions from each of the architectures. The weights of each pre-trained model have been fixed to 0.5 for the ResNet50-based model and 0.1 for the InceptionV3 and Xception-based models. The comparison between the results is visualized in Figure 9.

From this graph it is possible to observe that all the models are approximately equally able to predict the bone age around the edge areas. However, the ResNet50-based model is the best in predicting also the central area of the bone age axis. This suggests that the architecture based on patches of the original radigraphs is better at generalizing over all bone ages with respect to the other models. Probably because it is more able to focus on local features of the images. In Table 6, the results of a statistical analysis are presented, including the calculation of Mean Squared Error, Root Mean Squared Error and P-value between the predictions of each model and their respective ground truth:

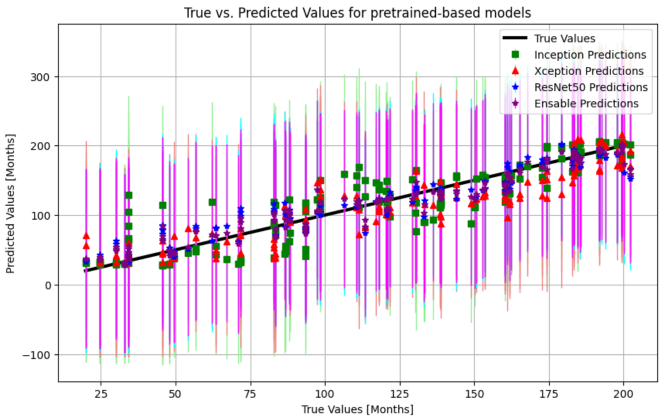


Fig. 9: Plot of the pre-trained models' predictions over the test set.

Model	MSE	RMSE	P-value
InceptionV3	0.19	0.44	0.89
Xception	0.18	0.42	0.91
ResNet50	0.06	0.24	0.97

TABLE 6: Comparison of different architecture's parameters.

In the predictive graph generated by AG-CNN, in Figure 10, we observe distinct behaviours among the three Branches. Notably, Global and Fusion Branches exhibit a consistent pattern in predicting the expected age with a narrow standard deviation, indicative of their robust performance. Conversely, the Local Branch displays a wider standard deviation, aligning with prior findings. This divergence is mainly attributable to the Local Branch's tendency to prioritize background lettering over salient image features of interest. Hence, while Global and Fusion Branches demonstrate similar predictive capabilities, the Local Branch drives away from its focus, resulting in increased variability in age predictions. In Table 7, the results of a statistical analysis are presented. Although these results are not as good as the ones obtained with the pre-trained models, the Global Branch alone seems to reach similar performances as the InceptionV3-based model. However, the results of the Fusion Branch are not as satisfactory as expected, since its predictions proved to be worse on the test set, compared to the ones of the Global Branch. This means that the Global Branch alone still performs better than the whole AG-CNN architecture.

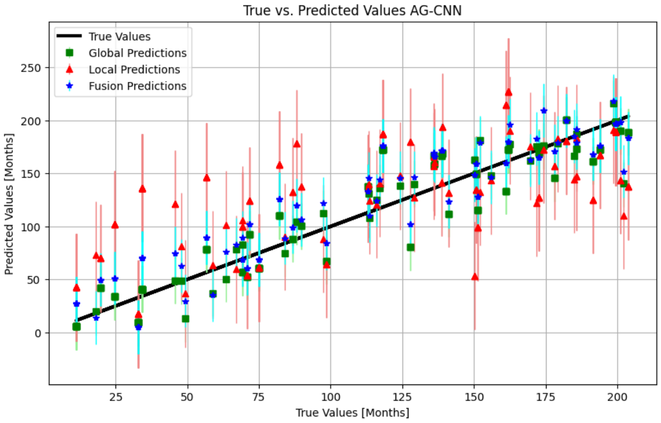


Fig. 10: Plot of AG-CNN's predictions over the test set.

Model branch	MSE	RMSE	P-value
Global	0.18	0.43	0.91
Local	0.79	0.89	0.57
Fusion	0.22	0.47	0.89

TABLE 7: Comparison of different branches's parameters.

VII. Concluding Remarks

In conclusion, the replication of the original winning model from RSNA bone age challenge was successfully achieved. The introduction of a novel architecture using Xception-based model was crucial to improve the overall performances and reduce computational costs. Additional improvements have been accomplished with the introduction of the patch-based approach which outperformed all models both in the stability and generalization of predictions.

The implementation of the visual attention mechanism through the AG-CNN architecture has been accomplished, although not as satisfactory as we expected. Further improvements should be taken into consideration in order to increase the performance of the model, starting from the exclusion of background elements from the attention images. Other improvements should involve also a longer training process, together with the reduction of input image dimensions for a significant reduction of computation times at the expenses of slightly less accurate predictions. Furthermore, a different way of computing the gradients, involving the general loss $\mathcal{L}$ with respect to the whole ensemble of trainable variables of the architecture may bring additional improvements to the performances.

Further investigations into above proposed solutions, could potentially bridge the performance gap between attention and patch-based approaches. Finally, also a general fine-tuning, in particular regarding the learning rate, for all proposed models could promise better performances.

References

[1] T. Widek, P. Genet, T. Ehammer, T. Schwark, M. Urschler, and E. Scheurer, "Bone age estimation with the greulich-pyle atlas using 3t mr images of hand and wrist," *Forensic Science International*, vol. 319, p. 110654, 2021.

[2] A. K. Poznanski, "Assessment of skeletal maturity and prediction of adult height (tw2 method)," *American Journal of Diseases of Children*, vol. 131, no. 9, pp. 1041–1042, 1977.

[3] A. Rakhlin, A. Shvets, V. Iglovikov, and A. A. Kalinin, "Deep convolutional neural networks for breast cancer histology image analysis," in *Image Analysis and Recognition: 15th International Conference, ICIAR 2018, Póvoa de Varzim, Portugal, June 27–29, 2018, Proceedings 15*, pp. 737–744, Springer, 2018.

[4] A. Tiulpin, J. Thevenot, E. Rahtu, P. Lehenkari, and S. Saarakkala, "Automatic knee osteoarthritis diagnosis from plain radiographs: a deep learning-based approach," *Scientific reports*, vol. 8, no. 1, p. 1727, 2018.

[5] 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.

[6] S. S. Halabi, L. M. Prevedello, J. Kalpathy-Cramer, A. B. Mamonov, A. Bilbily, M. Cicero, I. Pan, L. A. Pereira, R. T. Sousa, N. Abdala, *et al.*, "The rsna pediatric bone age machine learning challenge," *Radiology*, vol. 290, no. 2, pp. 498–503, 2019.

[7] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016.

[8] Q. Guan, Y. Huang, Z. Zhong, Z. Zheng, L. Zheng, and Y. Yang, "Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification," 2018.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.