Automated Bone Age Classification with Deep Neural Networks

Matthew Chen
Stanford University
mcc17@stanford.edu

Abstract

In this paper we look at the use of Convolutional Neural Network methods to train a model to predict developmental bone age of a patient given x-ray images. We use the Digital Hand Atlas dataset which is composed of scans independently annotated by two radiologists. Previous methods for this task generally involve a pipeline of segmentation and hand crafted feature extraction. We look to move away from this approach given recent advances in the effectiveness of convolutional neural networks for image classification.

We find that using a convolutional neural network approach for this image classification task, we are able to achieve a top one and two accuracy of 46% and 70% respectively with root mean squared error of 1.1 years, on our validation set. We observe that our largest jump in accuracy resulted from augmenting our dataset with random distortions. This seems to indicate that the performance is largely dependent on the number of training examples and would likely see further improvement with more data.

1. Introduction

Bone age assessment is a standardized process by which a medical practitioner determines the skeletal maturity of a child through a scan of their hand. Due to the nature of skeletal growth, this test is only accurate between the ages of 0 to 19. It is commonly used in comparison with chronological age as an indicator for developmental issues for a child. It is also useful in determining age where birth records are not accessible. This is particularly important in many parts of the world where most births are not recorded and accurate age estimates are needed later in life for events such as immigration and sporting [7].

The standard test for bone age assessment involves a radiological scan of the left hand which is then manually compared to a atlas of reference images. In this manual method the radiologist generally looks for certain characteristics of the image in regions of interest and either gives a holistic assessment as in the GP method [5], or gives a assessment which is a function of sub-scores given for specific parts of the image, as in the TW2 method [14].

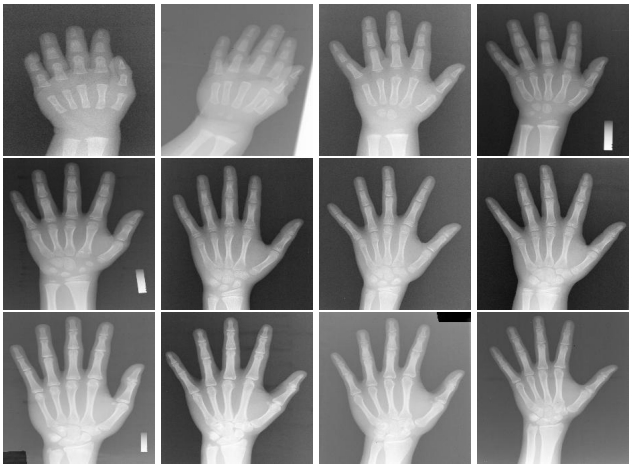

Figure 1. Example left hand scans across various ages from the digital hand atlas dataset

Automated methods for bone age assessment have been proposed in the past. These methods generally involve segmenting the scan into regions of interest and running a classifier on the results. In this paper we aim for a more general approach where we avoid creating hand crafted features by training a convolutional neural network directly on the input pixels.

2. Problem Statement

We formulate the bone age assessment problem as a classification problem where we receive left hand radiological scans as input and output a class corresponding to age as output. We will use the digital hand atlas dataset, a dataset created by the Childrens Hospital Los Angeles, funded by the NIH, composed of around 1400 hand scans across 19 years of ages and stages of bone development [4].

Examples of images from the dataset are shown in Figure 1. Each image is associated with the child's chronological age as well as annotated with developmental age assessments by two independent radiologists. The dataset also provides a racial and gender breakdown of the subjects. While there is evidence that these dimensions influence the

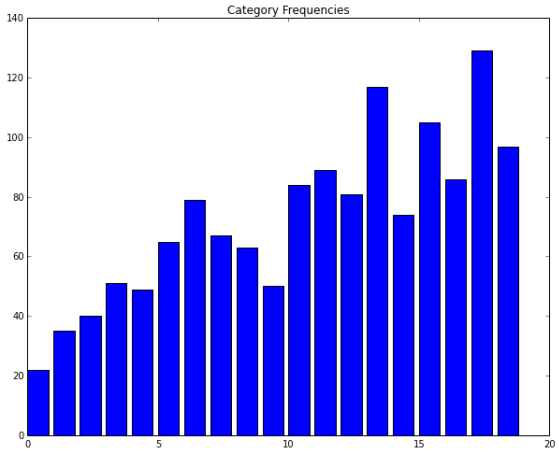

Figure 2. Age Ground Truth Distribution

development of skeletal structure, we do not use them in our model as it would greatly reduce the number of cases for each class in our already limited dataset. For the ground truth labels we will use the floor of the average of the two radiologist ratings for the image. The reason we use the radiologist's assessment of the childs bone age instead of their actual chronological age is that our aim is to replicate the process of assessing developmental age, not chronological age. Figure 2 shows the distribution of ground truth age labels for our dataset.

We later evaluate our results by measuring the accuracy of our classifier on a validation set. To get this split we partition our data into a training set and a validation set. The training data contains roughly 75% of the original data (around 1,000 images) while the validation set contains the rest. The split was done randomly, while roughly keeping the same portions of each class in each set. In addition to accuracy we will also look at the confusion matrix generated by our classifier to see if the errors are reasonable in terms of getting close to correct age category. To measure this we will look at the root mean square error (RMSE) of our classifier.

3. Related Work

Many classic computer vision and image processing techniques have been applied to the general problem of medical image classification with varying success. In the past, these techniques have generally involved some assortment of segmentation, frequently background subtraction, and hand crafted features with a learning model on top. While these methods have achieved reasonable accuracies in many domains, they generally require very specific hand engineered features to work, which greatly diminishes their ability to generalize to related problems.

Similarly for the problem of bone age assessment, many of the best methods involve segmenting the image into regions of interest and creating custom descriptors for these regions [2]. In one example, a method was created to segment out the region representing the Carpal Bone which was determined to be a good predictor of age for children under 7 [17]. The method performed well for ages below 5 (80% - 100%), however the performance fell off after such point. Another system segments the different bones in the image and measures their distances, followed by a lookup on a standardized chart which gave the relation of length ratios to bone age [8]. They compare their predictions with chronological age and get a mean difference of 1.57 years with 6% of the images rejected due to segmentation issues.

A popular commercial product for bone age assessment, called BoneXpert, is able to obtain 0.72 RMSE [15]. They use an active appearance model (AAM) to statistically match images on the grounds of shape and appearance parameters. As pointed out in [10], however their method relies on a relationship between bone age and chronological age, which is used as an input. The method also cannot handle too much noise in the image, rejecting a small percentage of input. In [10], they generate a continuous distribution of synthetically generated images by using software to interpolate between different aged scans. They then establish a method which uses a combination of SIFT features and SVD to create an image descriptor which is then used to train a fully connected neural network.

However this method, along with other similar ones, are not robust to images which may deviate from their internal models. This would include images which have high noise and ones which are misaligned from the standard template. Due to the varying datasets used and limited details released regarding these approaches we were not able to benchmark their performance on our dataset.

More recently, motivated by the success of deep learning techniques in general image classification [9][12][13], researchers have been exploring such methods on medical imaging. Researchers have applied the CNNs and variants to areas such as classifying patches of a lung CT scans to detect interstitial lung disease using a shallow network convolutional network [6]. In another example a pre-trained convolutional network is used as a feature extractor to classify chest x-ray pathology [1]. In many of these applications convolutional networks have been comparable or outperform previous state of the art results.

One issue that is more relevant to the medical image domain is collecting an annotated dataset of sufficient size to train a neural network. The impact of the size of the dataset on performance is examined in [3]. There they are able to achieve very high accuracy ($> 99\%$) for a body part classification problem with 200 cases. Our dataset involves the classification of images, which many be more ambiguous and granular in nature, however this number provides some reassurances in terms of the magnitude of images required

Section	Type	Parameters
Layer 1	conv3-64	1,728
	conv3-64	36,864
	maxpool	0
Layer 2	conv3-128	73,728
	conv3-128	147,456
	maxpool	0
Layer 3	conv3-256	294,912
	conv3-256	589,824
	conv3-256	589,824
	maxpool	0
Layer 4	conv3-512	1,179,648
	conv3-512	2,359,296
	conv3-512	2,359,296
	maxpool	0
Layer 5	conv3-512	2,359,296
	conv3-512	2,359,296
	conv3-512	2,359,296
	maxpool	0
Layer 6	fc-4096	102,760,448
Layer 7	fc-4096	16,777,216
	fc-19	77,824
	softmax	0
Total		134,325,952

Table 1. VGGNet architecture and number of parameters

to train a high accuracy classifier.

4. Approach

Given this image classification problem we apply different convolutional neural network architectures to train a classifier from the raw input pixels of the image. We use VGGNet as our baseline model. This network was one of the winners of the ImageNet challenge in 2014. To mitigate the issues with having a relatively small dataset, we use data augmentation methods such as random flips, rotations, and cropping to synthetically increase our training set size. We also use pretrained weights from models trained on large image repositories such as imagenet to initialize our weight parameters and fine tune our network with these weights. Finally we modify our loss function to take advantage of the fact that good predictions should be close together in range.

4.1. Pretrained Weights

We use VGGNet, chosen for its simple architecture, as our base model for this task. The fully architecture details along with the number of parameters at each layer are shown in Table 1. We retrieved the pretrained weights from Caffe Zoo, an online repository for pretrained network data, and convert the data format for our use. When using these

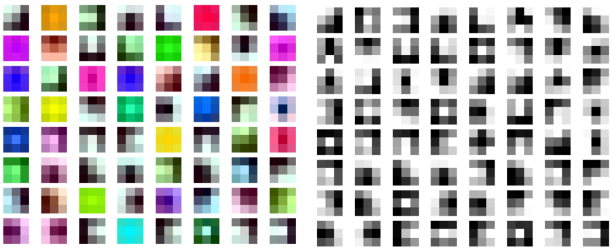

Figure 3. VGG16 RGB filters to Greyscale

pretrained weights from VGGNet, along with most other publicly available models, we need to make a decision in how to convert the input filter layer dimensions from RGB to grayscale. The reason for this is that imagenet is composed of RGB images while our radiological scans, and similarly most medical images in the standard DICOM format, is composed of grey scale equivalent images.

We make this conversion by taking the mean across the three RGB channels to generate new filter weights which can be used on single channel grey scale images. This decision seems reasonable as the process of converting a RGB image to grey scale would involve the same process. Additionally if we think that the filters represent edges, these edges are also apparent in the grey scale version of these filters as shown in Figure 3. Thus, at least in theory, they should hold the same semantic meaning as the original filters for use in the rest of the network.

4.2. Data Augmentation

Before we feed a batch of training data to our network we perform several preprocessing steps, which allow us to artificially increase the size of the training set. We do this by adding random distortions to the images during training time, which do not materially change the correct label of the image. First we perform mean subtraction across all images. This involves subtracting a single mean pixel value calculated for each image from all pixel values of that image. We also deploy random crops, which have a ratio of 0.875 to the original downsized image. Since our input for the various networks are 224 x 224 x 1 in size the original image size prior to being cropped is warped to 256 x 256 x 1. During test time we take six fixed size crops, four corners and two center, and average the scores of these six crops to get the resulting prediction. We also deploy random left right image flips and random rotations in the range of [-20,20] degrees.

Table 2 shows the number of unique images which are generated from a combination of the distortions in the training pipeline. We can see that the random distortions increase the synthetic training size by several orders of magnitude. While these examples are highly dependent, they help increase the network's robustness to translations and

Processing	Multiplier	Effective Images
Original	1	1,028
Random Rotate	41	42,148
Random Crop	1024	43,159,552
Random Flip	2	86,319,104

Table 2. Synthetic training examples

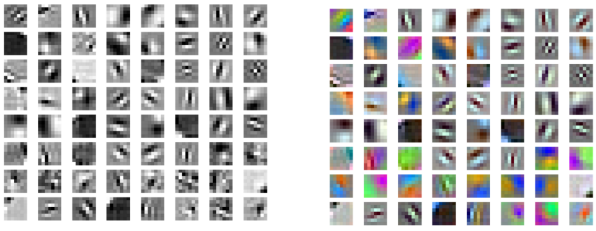

Figure 4. First convolution filters for Googlenet

rotations in the input image.

4.3. Loss Function

$$p_{ij} = \frac{e^{x_{ij}}}{\sum_k e^{x_{ik}}}$$

$$loss = \frac{1}{M}\sum_{i=1}^{M} -log(p_{y_i}) \quad \text{Cross Entropy}$$

Since we have a classification problem we use the softmax function as our final network output. The above equations show the calculation of the probabilities from the unnormalize scores using the softmax function and the associated cross entropy loss. Since cross entropy loss only takes into account the correct class when computing gradients, we sought to augment the loss function to increase the penalty for predicting ages which were further away from the true image age compared with ones that were closer. Thus we added a L2 loss function which is specified below.

$$loss = \frac{1}{M}(\sum_{i=1}^{M} -log(p_{y_i}) + \sum_{j=1}^{N} p_{i,j} * (j - y_i)^2) \quad \text{L2CE}$$

The first part is the standard cross entropy loss and the second part is the augmented L2 loss. We weight the L2 loss by the predicted probability. We found that empirically this loss performed better then either the cross entropy or L2 loss alone.

4.4. Architecture

We test the performance of various architectures on our dataset and compare our results. This includes our baseline VGGNet model [12], and GoogleNet [13]. The VGGNet architecture involves several repeating series of convolution, rectified linear units, and max pooling layers followed by fully connected layers. One negative aspect of VGGnet is the number of parameters in the model as shown in Table 1. We can see that most of these parameters occur at the first fully connected layer. We choose to compare the performance of this architecture with GoogleNet which had better performance on the Imagenet Challenge (6.7% vs 7.3% top 5 error) but had 12 times fewer parameters. For both architecture we initialize their parameters with pre-trained weight from imagenet. For Googlenet we use a similar process for the first convolutional layer where we average the first layer weights to get a single channel filter shown in Figure 4.

5. Results

In running our experiments we Adam as our optimization algorithm. We use an annealed learning rate strategy as input to our optimizer. We start at a learning rate of 0.0001 and decay the learning rate by 0.5 every 500 iterations for 5,000 iterations. We start with a relatively low learning rate so that we do not completely override the pre-trained initialization on the first few steps. Each training step uses a batch size of 32 due to memory constraints on the GPU that was used for the experiments. We trained each network for 5,000 iterations which equates to around 160 epochs of our training data.

5.1. Evaluation

To evaluate our results we look at two main metrics. The first is top 1 and 2 accuracy which shows the first and second top predictions of our network and compares it to the ground truth data. In our case the top 2 accuracy is particularly relevant due to the discretization step in our method where we defined the ground truth to be the floor of the average rating from the two independent readings. Due to this step a 2.9 and 3.1 ground truth will be categorized as 2 and 3 respectively even though they only differed by 0.2 years. Hence using the top 2 predictions should alleviate some of the issues with the ambiguity in our encoding scheme. Our next metric is Root Mean Squared Error which measures, on average, how far off our results were from the true labels. The full formula for calculating this metric is shown below.

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\underset{j}{\text{argmax}}(p_{ij}) - y_i)^2}$$

Using these metrics, Table 3 shows a summary of the results of the various experiments which we ran. We start

Method	Top 1 Acc	Top 2 Acc	RMSE
Radiologist	0.66	0.97	0.65
BoneXpert	NA	NA	0.72
VGGNet Base	0.32	NA	2.28
VGGNet	0.46	0.70	1.11
GoogleNet	0.36	0.65	1.25

Table 3. Comparison of Methods

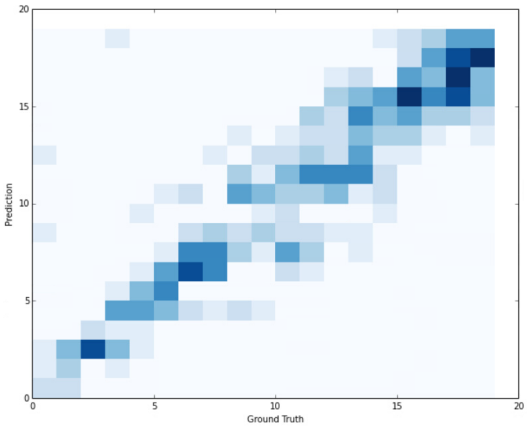

Figure 5. VGGNet base model confusion matrix

with VGGNet base which extracts image features by fixing the convolutional layer weights and tuning the fully connected layers. Then we augment the loss function with L2 loss and add data augmentation techniques to reach the final configuration for VGGNet and Googlenet which also used pre-trained weights as initialization but fine-tuned across all layers instead of just the fully connected ones.

5.2. Base Model

Using pre-trained weights from VGGNet [12] we fine tuned the fully connected layers, randomly initializing weights of the final output layer. For this baseline model we used only cross entropy loss without the L2 augmentation. Additionally we used a regularization parameter of 0.1 for the fully connected layers, which empirically was shown to be the most effective. We were able to obtain a top 1 accuracy of 32% and a RMSE of 2.28 years on the validation set. We can see this in the confusion matrix shown in Figure 5, as most of the predictions hover relatively close to the correct rating as shown on the diagonal.

5.3. L2 Loss

While analyzing the results of our baseline model we noticed some outliers in the predictions. This can be seen as outliers in the matrix in Figure 5. Since the cross entropy loss only provides information on the correct class the network is not given valuable information on the degree in which a prediction is incorrect. While this still produced

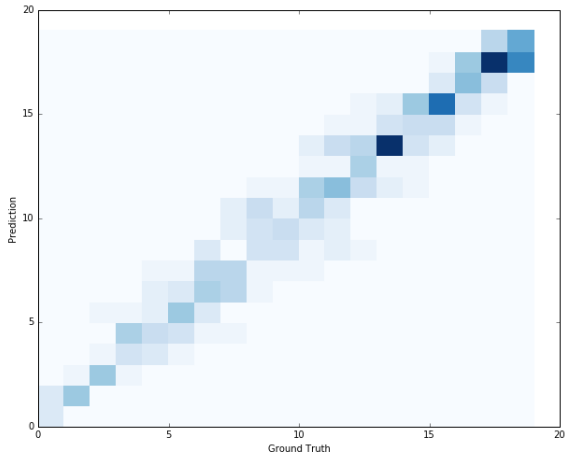

Figure 6. VGGNet with Data Augmentation Confusion Matrix

reasonable predictions for the most part, adding in L2 loss to cross entropy loss increased the performance of our baseline model to 36% top 1 accuracy and RMSE of 1.96 years.

5.4. Data Augmentation

During training of the baseline model, we noticed that without a high regularization parameter, the model quickly fit the training data perfectly. This indicated over fitting as the model performed poorly on the validation set. To mitigate the issue we augmented the data as described in the previous section. When this method was implemented to just fine tune the fully connected layers it had a negligible effect, with top 1 accuracy and RMSE remaining at 36% and 1.91 years respectively. However at this point the training accuracy dropped to below 65% at its highest from almost 100%. This implied that our model had high bias. To deal with the issue we fine tuned the whole network instead of just the last two fully connected layers. This lead to an increase in accuracy and decrease in RMSE to 46% and 1.11 years respectively.

We can see the difference qualitatively when comparing our baseline confusion matrix in Figure 5 with results from VGGNet with L2 loss and data augmentation shown in Figure 6. Similarly results from GoogleNet with the same configurations are shown in Figure 7. We can see that the predictions are now much closer to the ground truth labels. Additional we do not see large outliers in our results as we saw in the baseline method.

5.5. Error Analysis

In order to get a better understanding of where to look to improve our model we look at images in the validation set which our model classified incorrectly. Since VGGNet

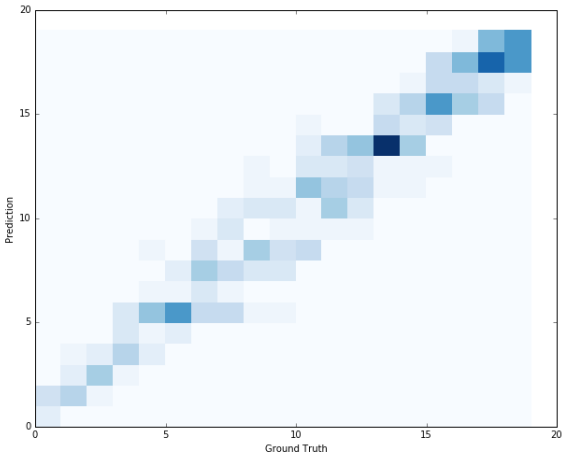

Figure 7. GoogleNet Net with Data Augmentation Confusion Matrix

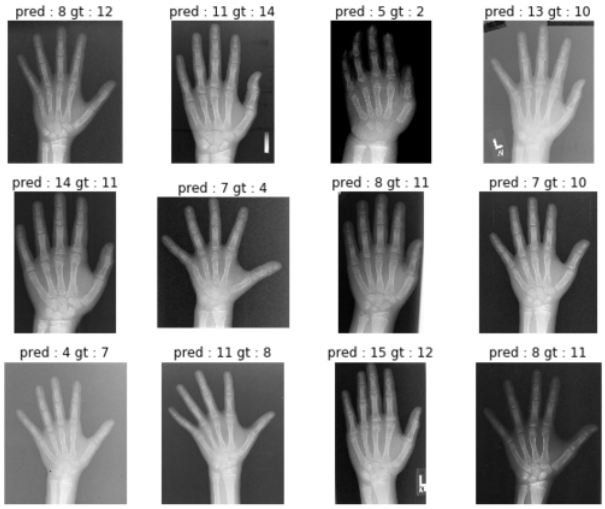

Figure 8. Incorrectly classified images ranging from highest to lowest squared error

was our best performing model we will use its classification errors for this analysis. Examples of misclassified images sorted by largest to smallest squared error is shown in Figure 8.

While looking at these images there are no very obvious patterns to the mistakes at first glance. One issue may be the contrast of the image which is fairly low in several of these cases. One way to mitigate the issue is to improve the contrast processing step in the training phase so that the network increases its invariance to contrast differences.

In Figure 9 we can see sample output from our classifier as it returns a probability distribution over the possible classes from the softmax function. In many cases even

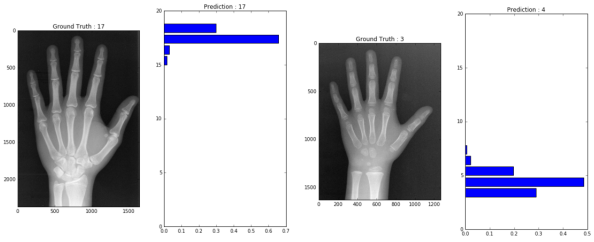

Figure 9. Sample output from classifier

when the classifier gets its top choice wrong the correct choice is within the range in which the network assigns high probability estimates. In these two cases the left one is correctly classified as age 17 and the right one was incorrectly classified as age 4 when with ground truth at age 3.

6. Conclusion

We have shown that our method of using a convolutional neural network to replace a pipeline of segmentation and classification has been successful in achieving results that are close to the current state of the art method of automated bone age assessment. We were able to achieve these results largely through the use of data augmentation techniques to artificially increase the size of our training set. This indicates that an increase in the original dataset would likely lead to an additional improvement in the accuracy of our classifier. Additionally, in our error analysis, we saw that there are additional types of invariance, contrast invariance in specific, which we can focus on to improve our overall accuracy.

Our current implementation treats the convolutional neural network classifier as a black box function which has been optimized for our task. For future work, in order to get a better understanding of the types of features which are being extracted from the image, we can use gradient based approaches such as [11] [16] to visualize what the network is learning. It would be especially interesting to see if the features which are extracted match up with the regions of interest which radiologists used to determine bone age from the GP and TW2 methods.

Additional future work can focus on transfer learning neural network weights from tasks which are related to the specific medical imaging task. For instance, in this case, another task involving classification of x-ray images. The motivation for this is that, while pre-training on ImageNet has proven to be effective, having low level features which are specific to medical images or even image modality specific may improve the accuracy of such classifiers. Evidence of this was seen in the increase in performance that was achieved when we moved from fine tuning the final two layers to tuning the whole network on our given task.

References

[1] Y. Anavi, I. Kogan, E. Gelbart, O. Geva, and H. Greenspan. A comparative study for chest radiograph image retrieval using binary texture and deep learning classification. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 2940–2943. IEEE, 2015.

[2] R. Bakthula and S. Agarwal. Automated human bone age assessment using image processing methods-survey. *International Journal of Computer Applications*, 104(13), 2014.

[3] J. Cho, K. Lee, E. Shin, G. Choy, and S. Do. Medical image deep learning with hospital pacs dataset. *arXiv preprint arXiv:1511.06348*, 2015.

[4] A. Gertych, A. Zhang, J. Sayre, S. Pospiech-Kurkowska, and H. Huang. Bone age assessment of children using a digital hand atlas. *Computerized Medical Imaging and Graphics*, 31(4):322–331, 2007.

[5] W. W. Greulich and S. I. Pyle. Radiographic atlas of skeletal development of the hand and wrist. *The American Journal of the Medical Sciences*, 238(3):393, 1959.

[6] K.-L. Hua, C.-H. Hsu, S. C. Hidayati, W.-H. Cheng, and Y.-J. Chen. Computer-aided classification of lung nodules on computed tomography images via deep learning technique. *OncoTargets and therapy*, 8, 2015.

[7] A. M. Mughal, N. Hassan, and A. Ahmed. Bone age assessment methods: A critical review. 2013.

[8] E. Pietka, M. F. McNitt-Gray, M. Kuo, and H. Huang. Computer-assisted phalangeal analysis in skeletal age assessment. *Medical Imaging, IEEE Transactions on*, 10(4):616–620, 1991.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[10] J. Seok, B. Hyun, J. Kasa-Vubu, and A. Girard. Automated classification system for bone age x-ray images. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 208–213. IEEE, 2012.

[11] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

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

[13] 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*, pages 1–9, 2015.

[14] J. M. Tanner, R. Whitehouse, W. Marshall, M. Healty, and H. Goldstein. Assessment of skeleton maturity and maturity and prediction of adult height (tw2 method). 1975.

[15] H. H. Thodberg, S. Kreiborg, A. Juul, and K. D. Pedersen. The bonexpert method for automated determination of skeletal maturity. *Medical Imaging, IEEE Transactions on*, 28(1):52–66, 2009.

[16] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[17] A. Zhang, A. Gertych, and B. J. Liu. Automatic bone age assessment for young children from newborn to 7-year-old using carpal bones. *Computerized Medical Imaging and Graphics*, 31(4):299–310, 2007.