Measuring Patellar Instability using Deep Learning Predicted Caton-Deschamps Index

Micael Tchapmi
Stanford University
mtchapmi@stanford.edu

Jacob Azoulay
Stanford University
jazoulay@stanford.edu

Daniel Lee
Stanford University
dklee@stanford.edu

Abstract

In this paper, we aim to predict patellar height parameters from lateral x-ray images using convolutional neural networks. Our dataset consists of 304 pre-treatment lateral x-rays. We explore various data augmentation techniques to increase the size of our training dataset. We explore various state of the art models for image recognition tasks, experiment with many combinations of hyperparameters, and provide a comparison of their performances. We obtain an average key point distance of ***2 pixels*** *on our test dataset and a CDI error of* ***0.19*** *after training with a VGG16 network, pre-trained on the ImageNet dataset. We also design a baseline model that is small enough so that it can train quickly and compare its performance to other state of the art networks on our dataset. Our work provides a solid base for predicting patellar instability using convolutional neural networks.*

1. Introduction

Patellar instability, one of the most common knee injuries among adolescents accounting for 3% of all knee injuries, is a disease where the patella bone dislodges partially (subluxation) or completely (dislocation) from the groove at the end of the femur, resulting in an unstable kneecap [7]. Patellar instability can be caused by a shallow groove, loose ligaments, extremely flexible joints, or more commonly among adolescents a sharp impact to the kneecap during a fall or sports injury. There are two types of treatment for patellar instability: nonoperative treatment, which is considered a more conservative treatment that involves physical therapy, and operative treatment, which involves surgery. A majority of first time subluxation or dislocations are treated conservatively. However, up to 44% of conservative treatments result in a recurrence of patellar instability [4]. Moreover, patients with a history of two or more dislocations have a 50% chance of a recurrence [30].

To evaluate patellar instability, a physical examination and imaging can be performed. When conducting imaging, there are several radiological parameters that can be used to determine patellar height and consequently patellar instability. The Caton-Deschamps Index (CDI), the metric of interest in this study, is measured from lateral x-ray imaging and can be used to determine the severity of the knee injury and inform treatment decisions, such as whether to conduct surgery and the type of surgery [1].

The CDI is defined as the ratio between (A) the distance between the anterior angle of the tibial plateau to the most inferior aspect of the patellar articular surface and (B) the patellar articular surface length. Normal range for the CDI is between 0.6 and 1.3. Patella alta, signified by a CDI greater than 1.3, is an abnormally high patella that can result in knee pain, patellar instability, and Osgood-Schlatter's disease. Patella baja, signified by a CDI less than 0.6, is an abnormally low patella that can result in knee pain as well and limited knee flexion [23]. The CDI can be calculated using three key points to compute the distances (A) and (B) as shown in Fig. 1.

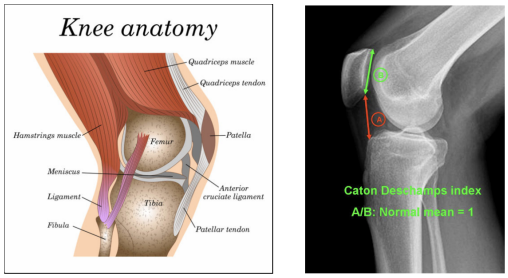

Figure 1. CDI calculation using three key points.

From the perspective of a radiologist, determining patellar height through the CDI from x-ray imaging is laborious, time consuming, and prone to significant inter-observer and intra-observer variability [3,27] Therefore, there is substantial motivation to develop computer based tools that can automatically extract the radiological parameters of interest from imaging, eliminating the effects of variability that originate from manual measurements by radiologists. One

example of a computer based tool is a deep learning algorithm such as a convolutional neural network (CNN).

2. Problem Statement

The project objective is to generate a CNN model that takes a lateral x-ray as an input and outputs the three key points that can be used to compute CDI. The model will be trained by a set of lateral x-rays with the three key points which have been hand labeled. Ideally, the model will output three key points that are identical to those identified by the radiologist. The overall approach taken for the problem was to

1. develop an infrastructure that can train a CNN and evaluate the CNN's performance on a validation set
2. investigate the performance of promising data transformations and CNN architectures
3. evaluate the performance of the best CNN model on the test set.

A two layer CNN was used as a baseline with its results described below.

3. Related Work

As described in a review article in Nature published in 2015, deep learning with the help of GPUs with fast growing compute power has enabled breakthroughs in numerous domains such as natural language processing and computer vision in a variety of industries such as pharmaceuticals to genomics [13]. Although deep learning has been popular for a few years, its use in medical imaging has only started accelerating in the last 5 years. The number of peer-reviewed publications, for instance, has grown from 100 in 2016 to more than 4000 in 2021 [25], many of which leverage state of the art CNN architectures, such as VGGNet or ResNet [8, 26], that perform extremely well on natural images such as those in ImageNet. Thus, deep learning in medical imaging is still nascent and prime for opportunity and exploration, especially in areas such as MRI processing, segmentation, and disease classification [17].

Although application of deep learning in medical imaging is an emerging field, there has been some notable work to date. Olczak *et al*. leveraged 5 popular deep learning networks (CaffeNet, VGG CNN S Network, VGG CNN 16, VGG CNN 19, and Network in network) to classify wrist, hand, and ankle radiographs into 4 classes (fracture, laterality, body part, and exam view) and demonstrated that their best network could perform at a human level [21]. For segmentation, Norman *et al*. demonstrated precision by segmenting meniscus from cartilage in the knee MR images with segmentations generated in 5 seconds on average [19]. This work ultimately speeds up the work flow to study knee degenerative diseases using MR imaging by leveraging a U-Net architecture, motivated by other successful segmentation applications of the U-Net on medical images for diseases such as pulmonary tuberculosis and brain tumors [5, 12, 16, 20, 24].

There are two methods to automatically measure radiological parameters related to patellar height and consequently patellar instability. The first involves leveraging deformable statistical models and the second involves leveraging deep learning techniques. Chen *et al*. automates measurement of the Insall-Salvati Ratio (ISR), another parameter used to gauge patellar instability, by combining a point distribution model, which captures the patella shape, with a canny edge detector to automate determination and calculation of the ISR [3]. Although this method successfully automates a task traditionally done manually by radiologists, it is slow when compared to employing a deep learning algorithm. It is evident when searching through literature that there has not been much work done in applying deep learning to determine patellar instability. However, there is one study that served as the primary reference, benchmark, and motivation for our work. Ye *et al*. leveraged a deep learning algorithm to automatically determine patellar height measurements [31] and as their work was published recently in 2020, we consider it state of the art and will compare our results to theirs throughout the report.

4. Data Processing

The data we will be using in this project comes from the JUPITER (Justifying Patellar Instability Treatment by Early Results) group. The data we will rely on is a subset from the Cincinnati Children's Hospital Medical Center, which includes x-ray images of the knees (either right or left) of patients aged 10-25 who have sustained patellar dislocation or subluxation. The data includes 304 pre-treatment lateral x-rays with key points manually labeled as ground truth and with the corresponding computed CDI. The key points are three sets of x and y pixel coordinates corresponding to the superior patella, inferior patella, and the tibial plateau.

First, the dicom medical image files are read and processed in Python using the pydicom library [18]. The data is pre-processed before being used by the neural network. All images are of various pixel dimensions varying from around 1500 pixels to 4500 pixels per side of the image. The data is composed of x-ray images which have one channel, making each image a two dimensional image. In order to standardize the data, each image is first center cropped to be a square of length equal to the shortest side length of the original image. The resulting cropped image is then rescaled to size 128x128 in order to standardize the data set to the same size. The data labels are also correspondingly scaled so key points remain in the same point on the knees in the x-rays.

Because the dataset is quite small, the model might be

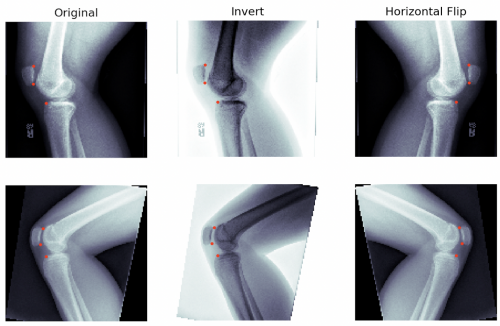

Figure 2. Inversion and horizontal flip augmentation methods on two sample x-rays.

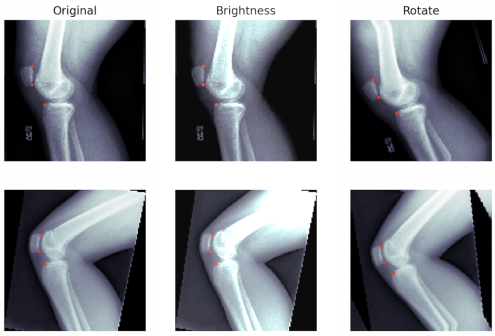

Figure 3. Brightness and rotation augmentation methods on two sample x-rays.

susceptible to overfitting. As a result, the data was augmented using various techniques. The Albumentation library in Python was used for all augmentations [9]. The center cropped and scaled data is used to create augmented data. The images are transformed by random rescaling, affine transformations, horizontal and vertical flips, rotations, color inversions, contrast adjustments, and brightness adjustments. These transformations were chosen because many of the original x-rays seem to vary by similar transformations. For instance, right knee x-rays can be flipped to create left knee x-rays. X-rays also tend to vary in scale, brightness, and contrast. Two sample images and corresponding augmentations are visualized in Fig. 2 and Fig. 3.

Other data pre-processing methods, such as only considering right knees, were experimented as well. While this would produce models that are not as generalizable, it serves as a performance comparison.

The data is split randomly into training, validation, and test sets with a 60/20/20% split respectively, which results in 182/61/61 images respectively) adhering to the constraint that all x-rays from each patient are all together in one of the three sets. Additionally, validation and test sets do not contain any augmented data.

After the data is segmented, all the images are normalized using the mean and standard deviation of features in the training dataset. Lastly, all label values, which are of value [0, 128], are rescaled to lie between [-1, 1].

An attempt to use Canny edge detection as a preprocessing method was also considered, which created a black background with white lines around prominent features as shown in Fig. 12 [2]. However, the model did not perform well likely because the majority of the pixels were black background pixels with a value of zero. Additionally, the labels which were originally made on the original images may not correspond to an edge on the edge image but rather a background pixel. One could automatically transfer the ground truth labels to the edge image by assigning a ground truth key point to the closest non-background pixel on the edge image but we did not explore this route further.

5. Methods

Given a lateral x-Ray, our goal is to measure patellar height using the CDI. To that effect, we train a CNN to predict 6 different coordinates (superior_patella_x, inferior_patella_x, tibial_plateau_x, superior_patella_y, inferior_patella_y, tibial_plateau_y) corresponding to three 2D points on our image. We then evaluate the network using the average CDI distance, average euclidean distance between the key points, and the Intra-Class Correlation coefficient (ICC). In this section, we describe our approach, including network architectures and metrics.

5.1. Architectures

We train and evaluate the performance of 5 different CNN's, including a baseline model, ResNet [8], VGG16 [26], U-Net [24] and AlexNet [11]. All implementation was done in Python using Pytorch [22].

5.1.1 Baseline Model

When deciding what network to train, we started by designing a simple baseline model, composed of 2 CONV-ReLU-MaxPool2D layers, followed by 2 Linear layers, each with a ReLu activation unit, and a final Linear layer at the output. The architecture is visualized in Fig. 4. We designed this network to be small enough so we could train quickly, easily test our training framework, and obtain preliminary results. We train our baseline model using the Mean Square error loss (MSE) which measures the distance between our predicted key points and the ground truth.

5.1.2 ResNet

ResNet-18 is an 18 layer deep convolutional network that is considered one of the primary models for medical image classification [8]. The network utilizes a residual learning framework to facilitate training as deep neural networks are

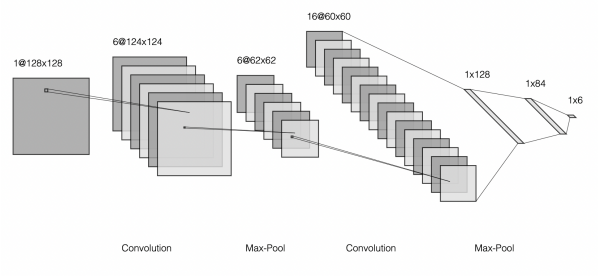

Figure 4. Baseline model architecture visualization [14].

difficult to train. ResNet is known for its expressibility due to its depth, which is important for many visual recognition tasks, and for winning 1st place in the ILSVRC 2015 classification task and 1st place in the ImageNet detection task. We employ this network by loading it pre-trained on ImageNet, changing the last linear layer to output the 6 key points of interest, and training it using MSE loss.

5.1.3 VGG16

VGG16 is a 16 layer deep CNN with about 138 million parameters [26]. Ye *et al*. [31] used an encoder-decoder network that uses skip connections for aggregating high and low-dimensional features to enhance the performance of their model, similar to the U-Net framework. They used VGG16, pre-trained on ImageNet, as the encoder network. In contrast to their work, we opted for a simpler setup where we also used VGG16 pre-trained on the ImageNet dataset but changed its output layer to a Linear layer that directly outputs our 6 key points of interest. As the network was pre-trained on RGB images, we duplicated our images across the RGB channels to make our input images compatible with the network. We then trained this network using the MSE loss.

5.1.4 U-Net

U-Net is a fully convolutional network widely known for biomedical image segmentation [24]. It uses an encoder-decoder architecture where the encoder learns an embedding (low dimensional representation of) the input image and the decoder learns to perform a task such as image segmentation from the embedding. Given that we don't have access to segmentation ground truth, we slightly modify this architecture so that it uses 2 decoders; one that learns to reconstruct the input image while the other learns to predict the key points from the embedding. We train this network end-to-end using a reconstruction loss and MSE loss. The reconstruction loss measures the distance between the input image and the reconstructed image.

5.1.5 AlexNet

AlexNet is a convolutional network with 60 million parameters and is widely known for winning the Imagenet large-scale visual recognition challenge in 2012 [11]. It consists of 5 convolutional layers, 3 max-pooling layers, 2 normalization layers, 2 fully connected layers, and 1 softmax layer, with each convolutional layer consisting of convolutional filters and a ReLU nonlinear activation function. We used an AlexNet model pre-trained on ImageNet and changed its output layer to a Linear layer that predicts our 6 key points of interest. We trained the network using the MSE loss.

5.2. Metrics

Closely following the work of Ye *et al*. [31], we report the performance of a trained network using 3 metrics: the CDI error, average key point distance, and the ICC coefficient.

5.2.1 Caton-Deschamps Index (CDI) Error

The CDI is a measure to determine the patellar height of a patient, which identifies patella alta or patella baja. The measurement is the ratio of the patellar articular surface length and the closest distance from the patella to the tibia [10]. The mathematical formulation is as follows:

$$tibial\_distance = \sqrt{(tp_x - ip_x)^2 + (tp_y - ip_y)^2} \quad (1)$$

$$patella\_articular = \sqrt{(sp_x - ip_x)^2 + (sp_y - ip_y)^2} \quad (2)$$

$$CDI = \frac{tibial\_distance}{patella\_articular} \quad (3)$$

where (tp_x, tp_y), (ip_x, ip_y), (sp_x, sp_y) are the x-y coordinates of the tibial plateau, interior patella, and superior patella respectively.

The CDI error measures how far the CDI of the input images are from their respective ground truth CDI. Since the CDI is a ratio, the CDI error is agnostic to image resolution or units of measurement.

5.2.2 Average Keypoint Distance

To determine the accuracy of the predicted key points, an average key point distance metric is calculated using the following formula:

$$\frac{1}{3}\sum_i \sqrt{(i_x^{gt} - i_x^{pred})^2 + (i_y^{gt} - i_y^{pred})^2}$$

$$\text{for } i \in [tp, ip, sp]. \quad (4)$$

The euclidean pixel distance from each predicted point to the corresponding ground truth point is calculated. The mean of these three pixel distance errors is then computed to produce the final average keypoint distance metric.

Because this metric is in units of pixels, the percent error can be calculated by dividing the result by the image side length, which in this case is 128 pixels. This percentage metric provides an error metric that is invariant to image resolution or scaling.

5.2.3 Intra-Class Correlation (ICC) Coefficient

The ICC coefficient is a statistical tool that measures the reliability of various agents to produce similar scores across different subjects. In the context of keypoint detection, the CDI determined from ground truth labels—which are manually identified by a human—will be compared with the CDI determined from the predicted labels—which are output by the model. The ICC takes into account variability across the two agents (the human and the model) and variability across the dataset.

The ICC can be mathematically defined using the following random effects model produced from two-factor ANOVA without replication:

$$x_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij} \tag{5}$$

The ICC is calculated using:

$$ICC = \frac{var(\beta)}{var(\alpha) + var(\beta) + var(\epsilon)} \tag{6}$$

Where $var(\beta)$ is variance due to differences across the dataset, $var(\epsilon)$ is the variance due to differences in CDI evaluations from the ground truth and predicted key points, and $var(\alpha)$ is the variance due to systematic differences between the CDI from the ground truth and predicted key points [15].

ICC values fall in the range [0.0, 1.0]. Values close to 1.0 indicate excellent reliability whereas values near 0.0 indicate low reliability. Negative ICC values are possible and can also be interpreted as having low reliability. A benchmark ICC between two human raters identifying CDI from x-ray images has been determined to be 0.66 [6]. All ICC calculations and implementations were conducted using the Pingouin Python library [29].

6. Experiments and Results

6.1. VGG16: Best Model Discussion

During experimentation, many different models were evaluated on the key point prediction task and their results were compared. Our VGG16 model outperformed all other models including our baseline model, U-Net, ResNet, and AlexNet. The VGG16 model also exhibited the least overfitting as shown by the loss plot in Fig. 5. See Appendix for the loss plots of the other models. Tab. 1 shows the average CDI distance, average key point distance, and average ICC evaluated on the test data set and as compared across all the models that were evaluated. Results from our primary reference (Ye *et al*. [31]), referred below as ModelY, are also included in Tab. 1 although it should be noted that we cannot directly compare our results to theirs as different data sets were used.

Model	CDI error	Key Point Distance	ICC
Baseline	0.42	8.68	-0.05
U-Net	0.25	4.5	-0.16
ResNet	0.21	3.14	0.22
AlexNet	0.22	4.77	0.26
VGG16	**0.19**	**2.05**	**0.33**
ModelY	0.06	0.06	0.91

Table 1. Comparison of model performances.

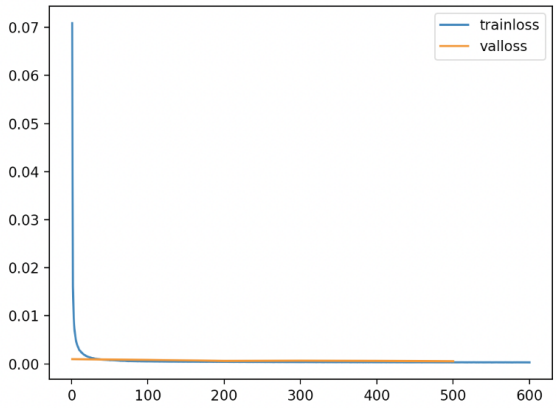

Figure 5. Train and validation losses (Y-axis) vs epoch number (X-axis) during VGG16 training.

After determining that the VGG16 model is superior to all the other models evaluated for the prediction task, we further tuned hyperparameters, predominantly batch size, learning rate, and selection of the optimizer. Initially, we started off with a larger batch size, a faster learning rate, and SGD as the optimizer. However, the best results that are obtained leverages a VGG16 model, a batch size of 8, a learning rate of 1e-5, 600 epochs, and an Adam optimizer. Fig. 6 shows a comparison of the best network's prediction with the ground truth key points.

Fig. 7 shows x-ray images that the VGG16 performed well on as seen by the ground truth key points overlapping with the model predictions. For these images, the average pixel errors are all around 0.5%. Note that the green dots

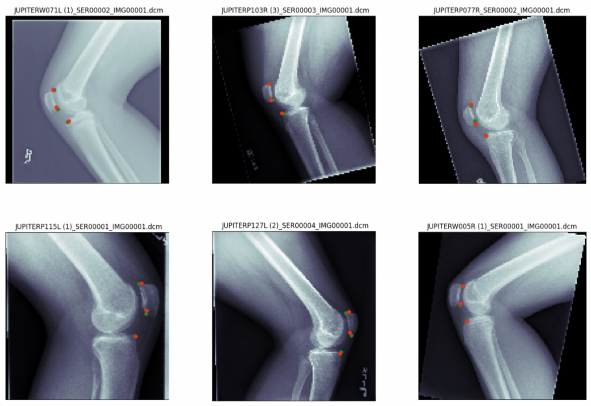

Figure 6. VGG16 keypoint predictions (red) vs Ground Truth key points (green).

are the ground truths and the red dots are the model's predictions. It's observed that all these x-ray success cases share the following commonalities: white tibia and femur overlaid on a black background, knee flexion, a clear and undistorted patella bone, and the image of the knee taken in a relatively up-right position.

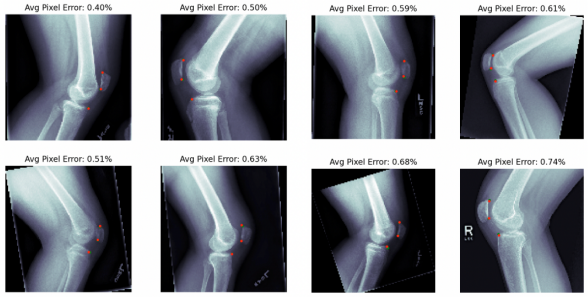

Figure 7. X-ray Success cases for VGG16.

Fig. 8 shows x-ray images that the VGG16 did not perform well, as evidenced by the model predictions being far from the ground truth key points and the average pixel errors ranging from 2% to 14%. It's worth noting that for the x-ray images that resulted in 2% average pixel error, the ground truth and the predictions do not visually appear that far apart and the model is still able to catch the patella bone in almost all of these cases. There's no apparent commonality among all the failure cases that can be discerned. However, the failure cases suggest the following possible failure modes: knee not bent, faint patella bone and/or minimal image contrast, and the knee rotated 90 degrees from the upright position. It's worth noting that if the x-ray images could be standardized to always look like the x-ray success cases, the model's performance would likely increase greatly.

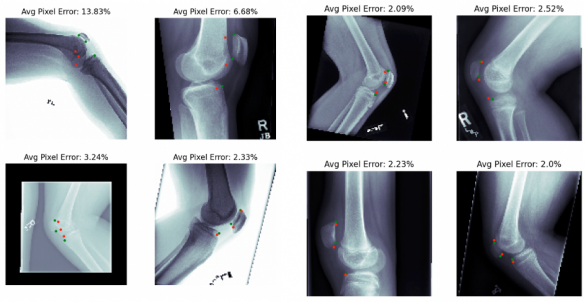

Figure 8. X-ray Failure cases for VGG16.

6.2. Saliency Map Visualization

To gain a better understanding of what parts of an image the CNN models are most influenced by, saliency maps can be produced. This visualization technique produces a heat map of the pixels that influence the final loss the most, measured by the absolute value of the gradients of each pixel input with respect to the loss.

Fig. 9 demonstrates a few sample x-rays and their corresponding saliency maps produced by the VGGNet. In almost all images, the regions near the patella have a significantly higher influence on the prediction of the model. This gives a good indication that the model is, in fact, not only able to identify where the patella is on the image but is also capable of using features of the patella to accurately predict the regions where the key points of interest are located. As a result, the model is more generalizable, as evidenced by the low overfitting.

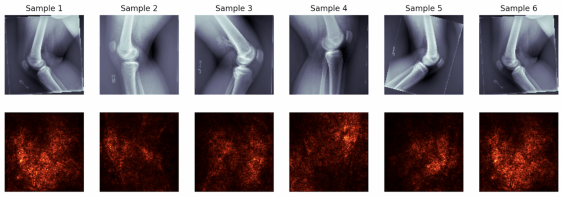

Figure 9. Saliency maps for select x-ray data samples.

It is interesting to note that the saliency maps also highlight regions of the tibia and femur, suggesting that the model's predictions are marginally influenced by the rest of the leg. Perhaps the model is better able to determine the patellar location by referencing the parts of the leg to which the patella is connected.

6.3. Feature Map Visualization

It is useful to visualize what the network is computing at each layer through the use of feature maps. This visualization technique gives insight into which features the model focuses on at each convolutional layer in the network. The final feature maps are computed by summing all output fil-

ters of a given convolutional layer [28]. Feature maps for the first 12 convolutional layers of the best performing VGGNet are displayed in Fig. 10.

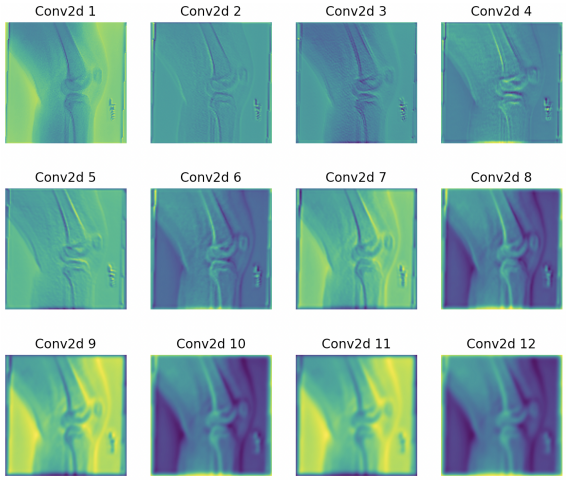

Figure 10. Feature map visualization of first 12 convolutional layers of pre-trained VGGNet.

From the feature maps, it is clear that the network is quite capable of identifying the tibia and femur, as evidenced by the significant contrast between the large bones and the surrounding flesh. The patella, however, may at times be less identifiable as seen in the 11th and 12th convolutional layers for instance. As with the saliency maps, the feature maps seem to suggest that the location of the tibia and femur can provide useful information when predicting the patellar key points of interest.

Lastly, a class visualization technique in which ground truth labels are provided and gradient descent is performed on a random noise image. The algorithm updates the random noise image to minimize its loss with respect to the provided labels. The goal is to generate an image that resembles the general structures of the x-ray; however, the results did not produce clearly discernible features as displayed in Fig. 16 in the appendix.

6.4. Pre-trained Versus Trained from Scratch

Starting with a pre-trained model rather than training from scratch is advantageous because it can enhance performance of a network on a data set, especially if the data set is small, and reduce the amount of time required to train. However, starting with a pre-trained model can also negatively bias or impact the model especially if the data that the model is pre-trained on shares little to no similarity with the current data. It can also negatively degrade the model performance if pre-training causes the model to get stuck at an unfavorable local minima during optimization.

To verify that starting with a VGG16 pre-trained model on the ImageNet dataset is not detrimental, we compare its performance with that of a new VGG16 trained from scratch. Although the train average key point distances for the pre-trained model and new model are similar at 1.4 and 1.6 respectively, the average keypoint distances of the pre-trained model at 2.05 is significantly lower than that of the new model at 3.47 on the test set. Thus it appears that starting with a pre-trained model enhanced performance and significantly reduced overfitting.

To understand the source of the performance enhancement, a feature map is generated for the first 12 convolutional layers of the new model Fig. 11 and compared to that of the best VGG model Fig. 10. It is observed that the feature maps for the new model have significantly less contrast and definition, especially for the 6th to 12th layers. Rather, it appears that the new model is overfitting the training data with the first two convolutional layers, while the other layers have less impact, highlighting the benefit of starting with a pre-trained model.

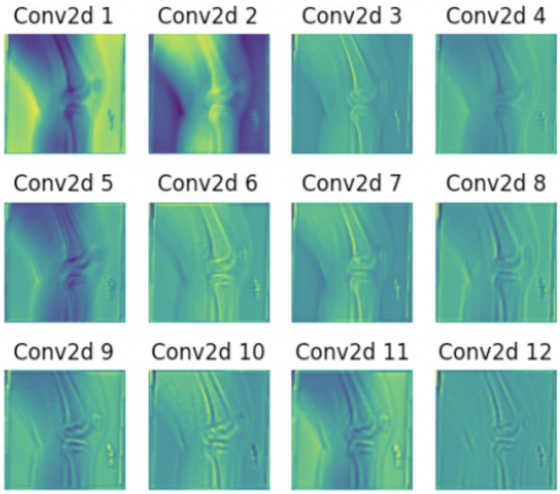

Figure 11. Feature map visualization of first 12 convolutional layers of VGGNet trained from scratch.

7. Conclusion

In a nascent field of applying deep learning to medical imaging, this project demonstrates that a VGG16 deep CNN can identify the 6 key points required on a knee x-ray to calculate the CDI even when trained on a small dataset with 304 x-ray images. The VGG16 outperforms our baseline model, U-Net, ResNet18, and Alexnet, which suggests that deeper networks perform better. This superior performance is obtained by leveraging data augmentations and starting with a pre-trained model on the ImageNet dataset. It's also verified that the pre-trained VGG16 outperforms a VGG16 trained from scratch. When visualizing the saliency and feature maps of the VGG16, it's observed that detecting the

patella plays a significant role in the model's performance while the location of the tibia and femur also contribute but to a lesser extent.

Recommended future work includes additional hyperparameter tuning such as varying even more the number of data augmentations and batch size and experimenting with additional neural network architectures such as a pre-trained U-Net or even deeper networks such as VGG19. Although not related to the model, we also recommend collecting more x-rays to train on and exploring if improving the quality of the x-rays can help improve the network's performance as well as generalization.

8. Contributions & Acknowledgements

This project was conducted under the guidance of Marissa Lee, a biomechanical engineering PhD candidate at Stanford University and member of the Neuromuscular Biomechanics Lab.

All authors of this paper received Collaborative Institutional Training Initiative (CITI) certification in order to obtain and view anonymized medical data. The paper was conducted in adherence to the ethical research standards outlined by the CITI program.

All code developed for this project can be found on GitHub here.

Micael Tchapmi produced much of the high level code base framework for training models. Jacob Azoulay developed the data processing code base. Daniel Lee created the baseline CNN model and integrated existing CNN models into the code base. All members contributed to the training of the models and the writing of the paper.

9. Appendix

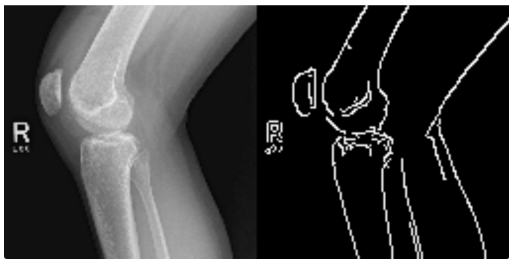

Figure 12. Augmentation methods on two sample x-rays.

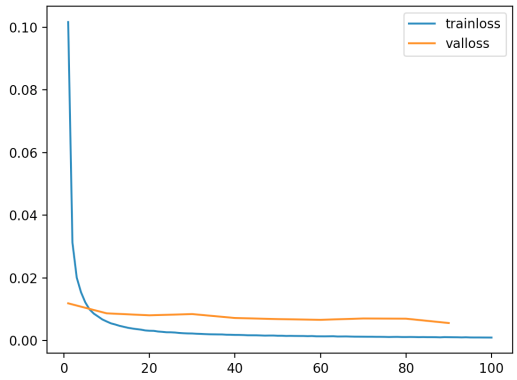

Figure 13. Train and validation losses (Y-axis) vs epoch number (X-axis) during AlexNet training.

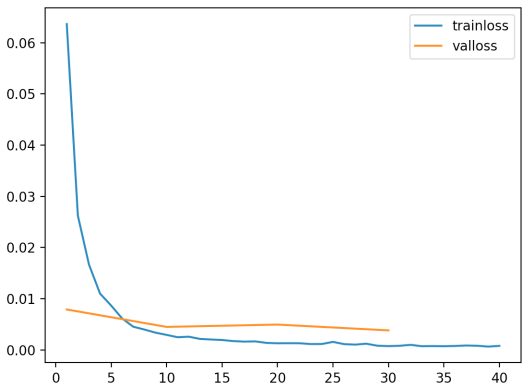

Figure 14. Train and validation losses (Y-axis) vs epoch number (X-axis) during U-Net training.

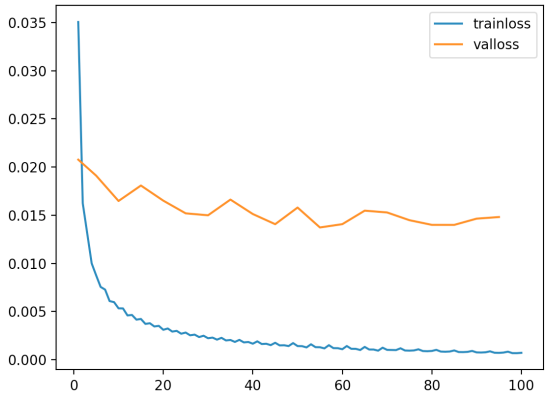

Figure 15. Train and validation losses (Y-axis) vs epoch number (X-axis) during training of baseline model.

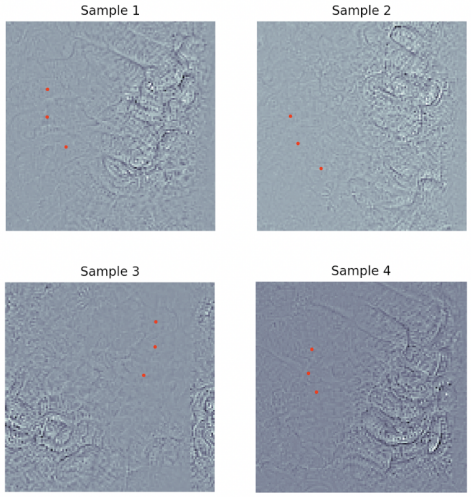

Figure 16. Loss minimizing generated images using VGG16.

References

[1] Massimo Berruto, Paolo Ferrua, Giulia Carimati, Francesco Uboldi, and Luca Gala. Patellofemoral instability: classification and imaging. *Joints*, 1(2):7, 2013. 1

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 3

[3] Hsin-Chen Chen, Chii-Jeng Lin, Chia-Hsing Wu, Chien-Kuo Wang, and Yung-Nien Sun. Automatic insall–salvati ratio measurement on lateral knee x-ray images using model-guided landmark localization. *Physics in Medicine & Biology*, 55(22):6785, 2010. 1, 2

[4] Riccardo D'Ambrosi, Katia Corona, Paolo Capitani, Gianluca Coccioli, Nicola Ursino, and Giuseppe Maria Peretti. Complications and recurrence of patellar instability after medial patellofemoral ligament reconstruction in children and adolescents: A systematic review. *Children*, 8(6):434, 2021. 1

[5] Bora Erden, Noah Gamboa, and Sam Wood. 3d convolutional neural network for brain tumor segmentation. *Computer Science, Stanford University, USA, Technical report*, 2017. 2

[6] Peter D Fabricant, Madison R Heath, Douglas N Mintz, Kathleen Emery, Matthew Veerkamp, Simone Gruber, Daniel W Green, Sabrina M Strickland, Eric J Wall, Beth E Shubin Stein, et al. Many radiographic and magnetic resonance imaging assessments for surgical decision making in pediatric patellofemoral instability patients demonstrate poor interrater reliability. *Arthroscopy: The Journal of Arthroscopic & Related Surgery*, 2022. 5

[7] Zara Hayat, Youssef El Bitar, and Justin L Case. Patella dislocation. In *StatPearls [Internet]*. StatPearls Publishing, 2022. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 3

[9] A. Buslaev A. Parinov E. Khvedchenya V.I. Iglovikov and A.A. Kalinin. Albumentations: fast and flexible image augmentations. *ArXiv e-prints*, 2018. 3

[10] Henry Knipe. Caton-deschamps index (knee): Radiology reference article, Jan 2022. 4

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 3, 4

[12] Paras Lakhani and Baskaran Sundaram. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*, 284(2):574–582, 2017. 2

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 2

[14] Alexander Lenail. Nn-svg. 4

[15] David Liljequist, Britt Elfving, and Kirsti Skavberg Roaldsen. Intraclass correlation–a discussion and demonstration of basic features. *PloS one*, 14(7):e0219854, 2019. 5

[16] Fang Liu, Zhaoye Zhou, Hyungseok Jang, Alexey Samsonov, Gengyan Zhao, and Richard Kijowski. Deep convolutional neural network and 3d deformable approach for tissue segmentation in musculoskeletal magnetic resonance imaging. *Magnetic resonance in medicine*, 79(4):2379–2391, 2018. 2

[17] Alexander Selvikvåg Lundervold and Arvid Lundervold. An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127, 2019. 2

[18] Darcy Mason. Su-e-t-33: pydicom: an open source dicom library. *Medical Physics*, 38(6Part10):3493–3493, 2011. 2

[19] Berk Norman, Valentina Pedoia, and Sharmila Majumdar. Use of 2d u-net convolutional neural networks for automated cartilage and meniscus segmentation of knee mr imaging data to determine relaxometry and morphometry. *Radiology*, 288(1):177–185, 2018. 2

[20] Alexey A Novikov, Dimitrios Lenis, David Major, Jiří Hladuvka, Maria Wimmer, and Katja Bühler. Fully convolutional architectures for multiclass segmentation in chest radiographs. *IEEE transactions on medical imaging*, 37(8):1865–1876, 2018. 2

[21] Jakub Olczak, Niklas Fahlberg, Atsuto Maki, Ali Sharif Razavian, Anthony Jilert, André Stark, Olof Sköldenberg, and Max Gordon. Artificial intelligence for analyzing orthopedic trauma radiographs: deep learning algorithms—are they on par with humans for diagnosing fractures? *Acta orthopaedica*, 88(6):581–586, 2017. 2

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 3

[23] CL Phillips, DAT Silver, PJ Schranz, and V Mandalia. The measurement of patellar height: a review of the methods of imaging. *The Journal of Bone and Joint Surgery. British volume*, 92(8):1045–1053, 2010. 1

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2, 3, 4

[25] Berkman Sahiner, Aria Pezeshk, Lubomir M Hadjiiski, Xiaosong Wang, Karen Drukker, Kenny H Cha, Ronald M Summers, and Maryellen L Giger. Deep learning in medical imaging and radiation therapy. *Medical physics*, 46(1):e1–e36, 2019. 2

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 3, 4

[27] Toby O Smith, Leigh Davies, Andoni P Toms, Caroline B Hing, and Simon T Donell. The reliability and validity of ra-

diological assessment for patellar instability. a systematic review and meta-analysis. *Skeletal radiology*, 40(4):399–414, 2011. 1

[28] Ravi vaishnav. Visualizing feature maps using pytorch, Jun 2021. 7

[29] Raphael Vallat. Pingouin: statistics in python. *The Journal of Open Source Software*, 3(31):1026, Nov. 2018. 5

[30] Steve Wolfe, Matthew Varacallo, Joshua D Thomas, Jeffrey J Carroll, and Chadi I Kahwaji. Patellar instability. *n/a*, 2018. 1

[31] Qin Ye, Qiang Shen, Wei Yang, Shuai Huang, Zhiqiang Jiang, Linyang He, and Xiangyang Gong. Development of automatic measurement for patellar height based on deep learning and knee radiographs. *European Radiology*, 30(9):4974–4984, 2020. 2, 4, 5