



Interpretable and Lightweight 3-D Deep Learning Model for Automated ACL Diagnosis

Young Seok Jeon , Kensuke Yoshino, Shigeo Hagiwara, Atsuya Watanabe, Swee Tian Quek, Hiroshi Yoshioka , and Mengling Feng, *Senior Member, IEEE*

Abstract—We propose an interpretable and lightweight 3D deep neural network model that diagnoses anterior cruciate ligament (ACL) tears from a knee MRI exam. Previous works focused primarily on achieving better diagnostic accuracy but paid less attention to practical aspects such as explainability and model size. They mainly relied on ImageNet pre-trained 2D deep neural network backbones, such as AlexNet or ResNet, which are computationally expensive. Some of them tried to interpret the models using post-inference visualization tools, such as CAM or Grad-CAM, which lack in generating accurate heatmaps. Our work addresses the two limitations by understanding the characteristics of ACL tear diagnosis. We argue that the semantic features required for classifying ACL tears are locally confined and highly homogeneous. We harness the unique characteristics of the task by incorporating: 1) attention modules and Gaussian positional encoding to reinforce the seeking of local features; 2) squeeze modules and fewer convolutional filters to reflect the homogeneity of the features. As a result, our model is interpretable: our attention modules can precisely highlight the ACL region without any location information given to them. Our model is extremely lightweight: consisting of only 43 K trainable parameters and 7.1 G of Floating-point operations per second (FLOPs), that is 225 times smaller and 91 times lesser than the previous state-of-the-art, respectively. Our model is accurate: our model outperforms the previous state-of-the-art with the average ROC-AUC of 0.983 and 0.980 on the Chiba and Stanford knee datasets, respectively.

Index Terms—ACL tear classification, interpretation, small deep neural network, 3D convolutional neural network.

I. INTRODUCTION

MRI is a widely accepted imaging technique for Anterior Cruciate Ligament (ACL) tear diagnosis. However, the current diagnosis process requires time-consuming manual examination by radiologists that is also error-prone at scale. A radiologist is required to examine each slice of the MRI scan, looking for ACL ruptures and other secondary complications, such as bone marrow edema and anterior tibial translation [3]. To improve the diagnosis productivity and accuracy, several AI models have been proposed to automate the ACL tear classification [1], [2], [4], [5]. Though the proposed models achieve good diagnostic accuracy, they suffer from two main limitations:

- 1) They employ Class Activation Map (CAM) to interpret learned features. However, the technique fails to isolate ACL from other neighboring ligaments and meniscus that are non-significant.
- 2) The models are heavily parameterized and computationally expensive. Thus, they are not suitable for resource-constrained devices and Federated Learning (FL).

A. Needs for Explainable Models

We need an interpretable AI for effective human-machine collaboration. The risk of model mis-classification is high in healthcare as it could directly affect a patient's well-being. To minimize the risk, we are required to understand the AI's decision-making process and discard if it is deemed illogical or does not coincide with radiologist's opinion [6].

The most common interpretation techniques adopted by state-of-the-art ACL tear classification models [1], [2] are CAM [7] and Grad-CAM [8], [9]. However, as shown in the bottom two rows of **Figure 1**, they produce imprecise ACL localization. The heatmaps only roughly highlight the central part of the knee that contains both ACL and other lesions (meniscus and posterior cruciate ligaments) that are non-significant. As a result, the heatmaps provide insufficient evidence for clinicians to accept the model's prediction. Furthermore, the techniques are computationally expensive. They are post-inference visualization tools that require additional computation overheads on top of model

Manuscript received August 26, 2020; revised April 9, 2021; accepted May 8, 2021. Date of publication May 18, 2021; date of current version July 20, 2021. This work was supported by the National Research Foundation Singapore under its AI Singapore Programme Award Numbers: AISG-GC-2019-001 and AISG-GC-2019-002, and in part by the NMRC Health Service Research under Grant MOH-000030-00. (*Corresponding author: Mengling Feng.*)

Young Seok Jeon and Mengling Feng are with the Saw Swee Hock School of Public Health and Institute of Data Science, National University of Singapore, Singapore 119077, Singapore (e-mail: youngseokjeon74@gmail.com; ephfm@nus.edu.sg).

Swee Tian Quek is with the Department of Diagnostic Imaging, at National University of Singapore, Singapore 119077, Singapore (e-mail: hiroshi@hs.uci.edu).

Kensuke Yoshino, Shigeo Hagiwara, and Atsuya Watanabe are with the Department of Orthopaedic Surgery, Chiba University, Chiba 263-8522, Japan (e-mail: atsuyan1@pa2.so-net.ne.jp; karakaze2000jp@yahoo.co.jp; knskysn@gmail.com).

Hiroshi Yoshioka is with the Department of Radiological Sciences, University of California Irvine, Irvine, CA 92697 USA (e-mail: quekswee_tian@hotmail.com).

Digital Object Identifier 10.1109/JBHI.2021.3081355

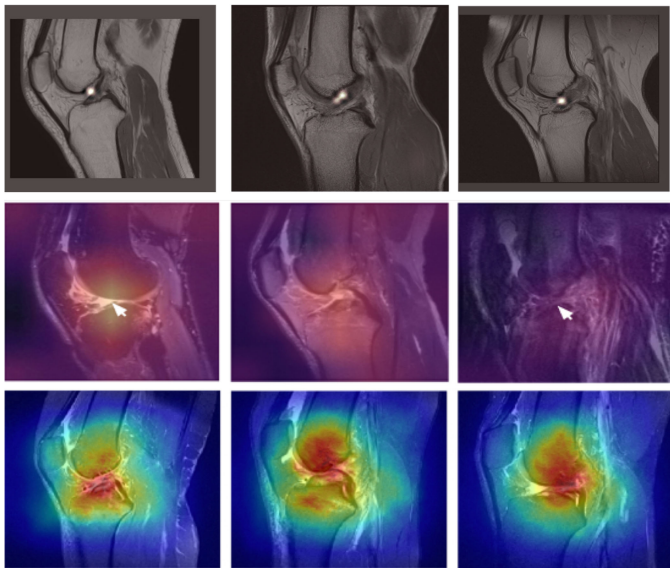


Fig. 1. Comparison of our proposed interpretation technique (top) to previously proposed techniques (middle and bottom). Our techniques achieves much sharper and accurate localization of torn ACL compared to previous state-of-the-arts: MRNet [1] (middle) and EL-Net [2] (bottom).

inference, making the techniques less suitable for practical deployments.

B. Needs for Smaller Models

Smaller deep neural networks (DNNs) [10]–[12] are preferred over bigger models if they have comparable performances. As illustrated in [Figure 2](#), smaller DNNs are not only suitable for resource-constrained devices but also accelerate Federated Learning (FL). They require much fewer memory and Floating-point operations per second (FLOPs); thus, they are more suitable for budget devices with limited computational power. Also, small DNNs solve the communication bandwidth problem of FL without having to compress the models [13], [14].

Despite the apparent benefits that small DNNs could offer, most AI applications in healthcare ignore the matter. With the ACL tear classification task, we show that it is not only possible but also more effective to build much smaller DNNs for medical imaging tasks.

C. Characteristics of ACL Tear Classification

Natural images are subjected to huge external variations such as lighting, viewpoint, and scale. Also, the object of interest can appear at any location. A strong DNN is a model that is robust to all of such variations. DNNs commonly solve the issue by introducing more parameters, as is evident from the ever-growing model size [11], [15]–[18]. With more parameters, DNN is capable of memorizing a wider range of variations [19], [20].

In contrast to the natural image tasks, ACL tear classification relies only on a handful of highly localized low-level features such as ACL fibers discontinuity, ACL angle, and bone marrow edema [3]. Also, Knee scans have a consistent viewpoint,

lighting, and scale. These imply that a small DNN is sufficient to memorize the small set of variations while not harming the model performance.

D. Contributions

We construct an interpretable and lightweight 3D Convolutional Neural Network (CNN) for ACL tear diagnosis based on our in-depth understanding of the task.

We introduce lightweight attention modules to visually interpret prediction outcomes instead of relying on post-inference visualization techniques such as CAM or Grad-CAM. Our attention module has several advantages over the post-inference visualization techniques. As aforementioned, features for ACL tear classification are highly localized. Our module is designed to focus only on locally confined features. Also, the technique introduces negligible computation overheads, making our technique more ideal for budget devices.

Also, we show that a small DNN is sufficient for achieving a near-human-level ACL detection performance. We minimize the model size by replacing standard convolutional modules to squeeze modules and using fewer convolutional filters. Our squeeze module, inspired by Fire module [11] and Ghost module [21], addresses the homogeneity characteristic of ACL tear classification task by re-using convolutional filters after transforming them with cheap linear operators.

II. RELATED WORKS

A. Explainable AI

Recent advancements in AI have led to widespread adoption of the technique in various industrial applications such as text translation [22], speech recognition [23] and recommendation system [24]. However, we observe a much slower adoption rate of AI from industries, such as healthcare, autonomous vehicle, and recruitment, that involve bigger risks or potentially raise ethical problems. For healthcare, in particular, we have a huge gap between the number of newly proposed techniques and their actual implementation cases. Explainability of AI models is regarded as the main reason for the slow adoption rate of AI [6]. Therefore, to accelerate the adoption rate of AI in healthcare, we not only have to focus on attaining better diagnostic accuracy but also improving its explainability [25].

1) *Explainable AI in Imaging Tasks*: In computer vision, given a grayscale image $\mathbf{x} \in \mathbb{R}^{H \times W}$, where H and W are the Heights and Width of the image, we interpret a CNN model f_θ by generating a saliency mask $\mathbf{a} \in [0, 1]^{H \times W}$. The saliency mask assigns higher values to regions with greater feature importance and vice versa. We group various saliency mask generation techniques into two: 1) model diagnosis and 2) attention approach. The model diagnosis approach does not require any modification to a CNN model. However, apart from the main classification task, it requires an additional computation in generating the saliency mask. On the other hand, the attention approach generates a saliency mask by embedding attention modules into a CNN model. The attention module is trained to deactivate non-significant regions in an image automatically.

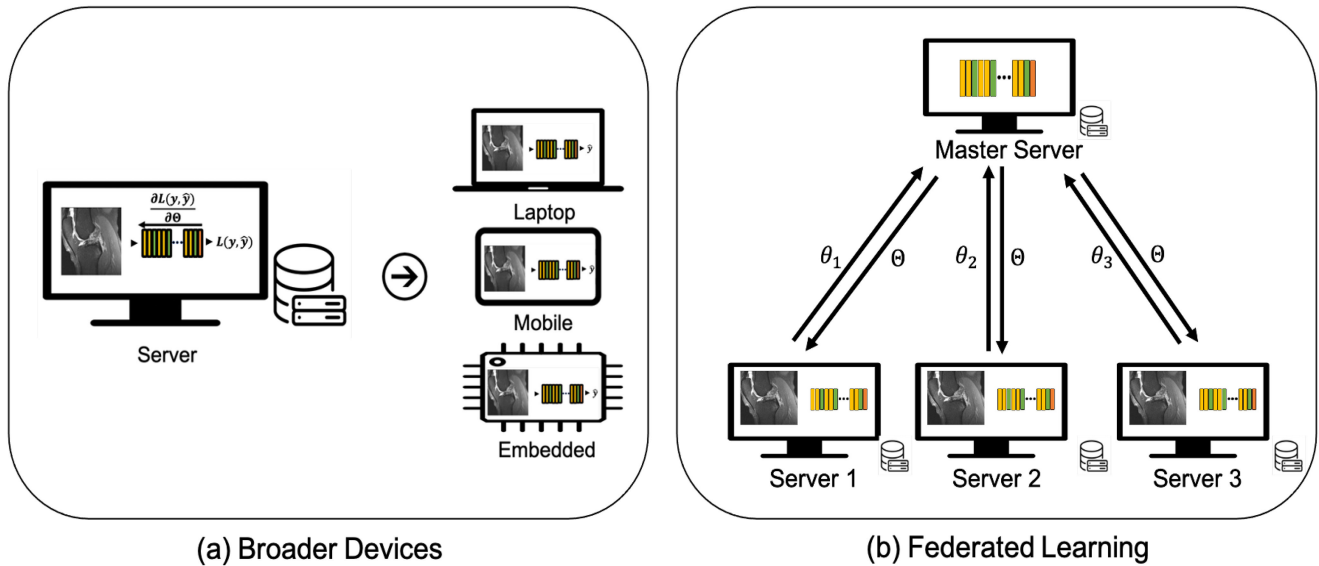


Fig. 2. Broader AI application with smaller DNNs. (a) Small deep neural nets (DNNs) integrate seamlessly to various devices, reducing the financial burden for health institutes in deploying AI systems. (b) Small DNNs accelerate Federated learning (FL) by minimizing the required communication bandwidth for parameter transfer.

2) Model Diagnosis Approach: One of the first attempts to interpret a CNN model via a saliency mask was proposed by Zeiler *et al.* [26]. The technique censors a small portion of an image \mathbf{x} with a binary mask $\mathbf{m}_{h,w}$, where h, w denote the center of masking region. Saliency mask is generated by mapping the classification accuracy measured across different masking locations (i.e. $\mathbf{a}_{h,w} = 1 - f_{\theta}(\mathbf{m}_{h,w} \odot \mathbf{x})$).

Local Interpretable Model-agnostic Explanations (LIME), proposed by Ribeiro *et al.* [27], produces non-rigid masking instances of an image with a super-pixel method. A saliency mask is generated by selecting the masking instance that produces the highest classification score.

CAM, proposed by Zhou *et al.* [7], generates the mask by computing the weighted sum of the final embedding \mathbf{z} with the weights from a linear projection head \mathbf{w} (i.e. $\mathbf{a}_{h,w} = \sum_k \mathbf{w} \cdot \mathbf{z}_{h,w}$), where k is the code size of the final CNN embedding \mathbf{z} . For Grad-CAM [8], [9], the weights \mathbf{w} are obtained by computing the derivatives of model prediction \hat{y} with respect to the embedding (i.e. $\mathbf{w} = \sum_{h,w} \frac{\partial \hat{y}}{\partial \mathbf{z}_{h,w}}$).

The model diagnosis approaches were used extensively in various medical imaging tasks. CAM and Grad-CAM were applied for interpreting chest radiograph diagnosis [28]–[30], bone fracture [31]–[33] and Alzheimer’s disease [34], [35]. LIME was applied for interpreting lymph node classification [36], as well as non-imaging tasks such as diabetes prediction [6].

3) Attention Approach: Wang *et al.* [37] proposed Residual Attention module. The proposed attention module is divided into two branches: mask branch and trunk branch. The mask branch is a small encoder-decoder architecture to produce an attention map with large receptive field size. The trunk branch performs a feature extraction that is indifferent to other CNN models.

Woo *et al.* [38] proposed Convolutional Block Attention Module (CBAM). CBAM has two modules that separately handle spatial and channel attentions. The spatial attention module is constructed similarly to the Residual Attention module. The

channel attention module is constructed with Squeeze-and-Excitation (SE) module [39].

Seo *et al.* [40] proposed progressive attention network (PAN). PAN introduces a query-guided attention module that suppresses irrelevant regions in an input image. However, the attention module is not applicable to a general classification task as it requires an additional query label. As a remedy to the problem, Jetley *et al.* [41] proposed a refined version of PAN that substitutes the query vector with a learnable representation.

For medical imaging, Oktay *et al.* [42] and Schlemper *et al.* [43] proposed attention gating (AG), a modified attention module from PAN, for fetal ultrasound screening and pancreas segmentation tasks. Hu *et al.* [44] modified CBAM for pediatric echocardiography segmentation task. Li *et al.* [45] applied attention module for grading cancer from a high-res whole slide image.

B. Small Models

AlexNet [46] revolutionized the way we view computer vision problems. Similar CNN models have been proposed ever since, mainly focusing on achieving better classification accuracy [11], [15]–[18]. However, the models have evolved to become more complex with newly proposed techniques and layers [16], [47], [48]. Simultaneously, there were attempts to oppose against the ever-growing complexity of CNN models. Cp-decomposition method, proposed by Lebedev *et al.* [49], reduces the computational cost of a 2D CNN model by decomposing a multi-channel 2D convolution operation into three separate 1D convolution operations. Forrest *et al.* [11] derived a CNN model that is 50 times smaller than AlexNet with Fire modules. Fire module computes the next hidden feature by first “squeezing” input features to a lower dimension and passes the squeezed feature to $[1 \times 1]$ and $[3 \times 3]$ convolutional layer simultaneously. The outputs from the two layers are then concatenated to form

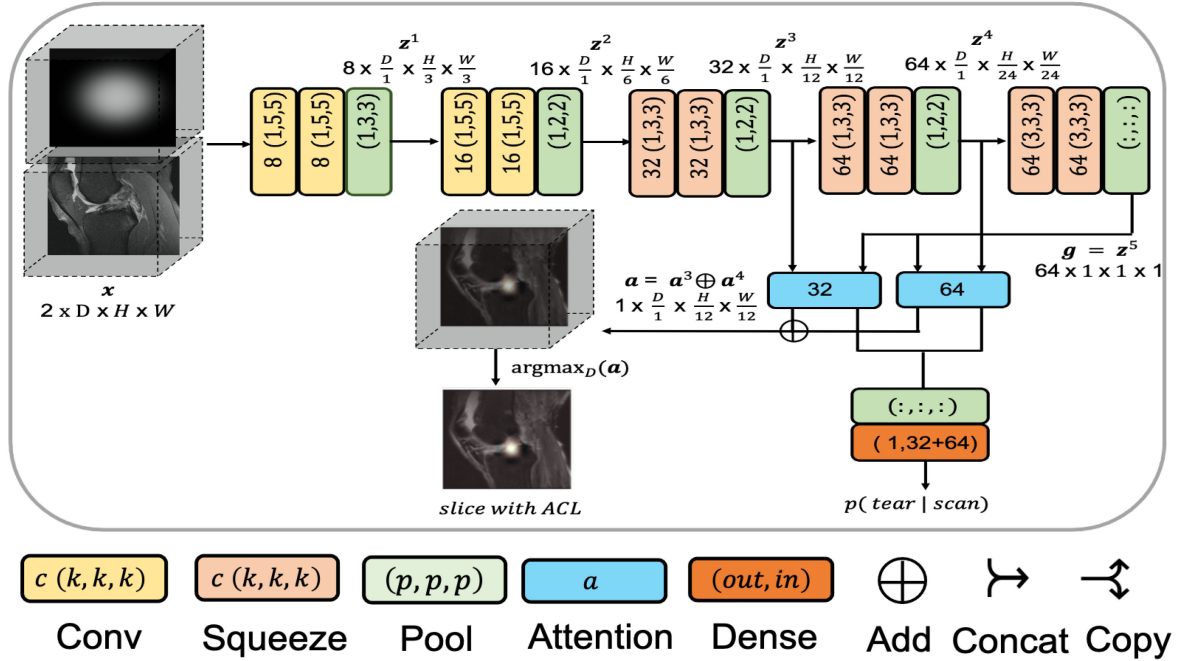


Fig. 3. Model Illustration. The two cubes on the left represent the inputs to our model: a 3D knee scan (bottom) and Gaussian feature (top), each with shape $[D \times H \times W]$, where D, H, W denotes the Depth, Height and Width of the scan. Our model outputs two values: an ACL tear prediction score $\mathbf{p}(\text{tear} | \text{scan})$ and a slice with highest attention score $\text{argmax}_D(\mathbf{a})$. Different computational units are represented with colored blocks. In the “Conv” and “Squeeze” units, c denotes the output channel size and (k, k, k) denotes the convolutional filter size. Though not explicitly mentioned, we introduce “ReLU” layer for every “conv” and “squeeze” units. In the “Attention” block, a denotes the number of $[1 \times 1 \times 1]$ convolution filters in the module. In the “Pool” block, (p, p, p) denotes pooling size. $(:, :, :)$ denotes a global max-pooling. Lastly, the (out, in) in the “Dense” layer denotes the size of output and input feature.

the final feature. ShuffleNet [10] proposed channel-wise Group convolution that splits input features into groups of features and applies convolution to each group. Channel shuffling is applied subsequently to promote the exchange of information across the groups. Han *et al.* [21] proposed a Ghost module that reuses convolutional filters by slightly transforming them with cheap operations.

C. Models for ACL Tear Detection

Bien *et al.* proposed MRNet [1]: a 2D CNN classification model that can predict three kinds of diagnosis (ACL tear, Meniscus tear, and Abnormality) from 3 different knee scans (Sagittal, Coronal, and Axial). The model predicts each diagnosis by separately encoding each plane with a 2D CNN model such as AlexNet and aggregating the encoded features with a linear classifier. The model has approximately **9 M parameters**.

Liu *et al.* [5] proposed a two-step approach that first segments cartilage lesions from a scan and finds diagnostic abnormalities from the segmented region. The segmentation is performed using 2D-UNet [50]. Small image patches are extracted around the segmented cartilage lesions. The extracted patches are used to fine-tune the encoder of 2D-UNet for ACL tear prediction. 2D-UNet is a huge architecture with **7 M parameters**.

Tsai *et al.* proposed ELNet [2]. The model achieves higher accuracy than MRNet by introducing two additional computational layers: Multi-Slice normalization and Blur-pool. The model currently achieves state-of-the-art accuracy in ACL detection with a ROC-AUC of 0.96 on the Stanford knee dataset. ELNet

is the smallest ACL tear classification model of all with **0.3 M parameters**.

III. METHODOLOGY

A. Architecture

As illustrated in Figure 3, our model takes two input features: a knee MRI scan and a Gaussian positional feature. Based on the two features, our model not only learns how to diagnose ACL tears but also highlights parts of the scan that contribute strongly to the diagnosis. Broadly, our model consists of 5 feature extraction modules, 2 attention modules that are branched out from the 3rd and 4th feature extraction modules, and a dense layer. The feature extraction modules progressively map the raw features (a knee MRI scan and a Gaussian positional feature) into a new set of features that represent more abstract meanings, such as ligaments, muscle, and bone. The attention modules compare the semantic similarity of the embedded features across different feature extraction modules. The dense layer maps the outputs from the two attention modules to a final diagnosis score. We provide a more detailed explanation of different learning blocks in the next sections.

1) Attention Module: Let’s denote the set of features extracted across L different feature extraction blocks by $\{\mathbf{z}^l\}_{l=1}^L$ and assume that the features have identical spatial size (i.e. $\mathbf{z}^l \in \mathbb{R}^{C^l \times D \times H \times W}$). Our attention module learns to compute the similarity $\mathbf{a}^l \in [0, 1]^{D \times H \times W}$ between the last feature $\mathbf{g} = \mathbf{z}^L$ and a feature from other layer \mathbf{z}^l , $l \neq L$:

$$\mathbf{a}_{d,h,w}^l = \langle \mathbf{z}_{d,h,w}^l, \mathbf{g}_{d,h,w} \rangle \quad (1)$$

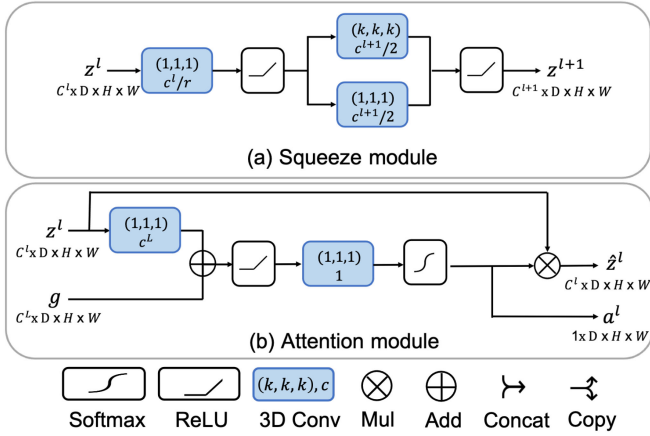


Fig. 4. Illustrations of (a) squeeze module and (b) attention module. (a) illustrates our proposed squeeze module that replaces the standard convolutional layer. Squeeze module computes the next hidden feature z^{l+1} by first squeezing an input feature z^l to a lower dimension using $[1 \times 1 \times 1]$ convolution layer with C^l/r filters. The resulting tensor passes through $[1 \times 1 \times 1]$ and $[k \times k \times k]$ convolutional layers, both with $C^{l+1}/2$ filters. Figure (b) illustrates our proposed attention modules that are mounted to the 3rd and 4th feature extraction blocks. The module generates an attention map by computing the similarity between two features: feature from the last layer g and any feature other previous layers z^l . We define the similarity as $\mathbf{a}_{d,h,w}^l = \sigma(\mathbf{w}_{\text{out}} \cdot \text{ReLU}(\mathbf{w}_{\text{in}}^T \mathbf{z}_{d,h,w}^l + \mathbf{g}_{d,h,w}))$, where \mathbf{w}_{in} and \mathbf{w}_{out} are the only learnable weights in the attention module.

We formulate the similarity function as :

$$\mathbf{a}_{d,h,w}^l = \sigma(\mathbf{w}_{\text{out}} \cdot \text{ReLU}(\mathbf{w}_{\text{in}}^T \mathbf{z}_{d,h,w}^l + \mathbf{g}_{d,h,w})) \quad (2)$$

ReLU is a non-linear activation function (i.e. $\text{ReLU}(x) = \max(0, x)$). We define the normalization function σ as the softmax across spatial dimensions (D, H, W) such that the attention values across the spatial dimensions sums to 1 (i.e. $\sum_{d,h,w} \mathbf{a}_{d,h,w}^l = 1$). This is to promote competition across the features in space such that it results in a more confined attention map. Our attention module is lightweight. $\mathbf{w}_{\text{in}} \in \mathbb{R}^{L \times l}$ and $\mathbf{w}_{\text{out}} \in \mathbb{R}^L$ are the only learnable weights in the module. A graphical illustration of our attention module is provided in Figure 4 (b). As shown in Figure 3, We apply the attention module to the outputs from the 3rd and 4th feature extraction block.

2) Squeeze Module: As shown in Figure 3, the first two feature extraction modules are typical convolutional blocks, each consisting of 2 3D convolutional layers (with Relu) and a pooling layer. For the remaining feature extraction modules, we replace convolutional layers with squeeze blocks to reduce our model's overall size.

Given a 3D feature $\mathbf{z}^l \in \mathbb{R}^{C^l \times D \times H \times W}$ in layer l , a standard convolutional layer computes the next hidden feature $\mathbf{z}^{l+1} \in \mathbb{R}^{C^{l+1} \times D \times H \times W}$ by convolving \mathbf{z}^l with C^{l+1} convolutional filters, each with $[k \times k \times k]$ kernel size. Therefore, a standard convolutional layer requires a total of $C^l \cdot C^{l+1} \cdot k^3$ trainable parameters.

As shown in Figure 4 (a), our squeeze module computes the next hidden feature \mathbf{z}^{l+1} by first “squeezeing” input features \mathbf{z}^l to a lower dimension using $[1 \times 1 \times 1]$ convolutional layer with

C^l/r filters. The constant r is a hyper-parameter that controls the degree of dimension reduction. The squeezed tensor then passes through $[1 \times 1 \times 1]$ and $[k \times k \times k]$ convolutional layers, both with $C^{l+1}/2$ filters. The final feature \mathbf{z}^{l+1} is computed by concatenating the outputs from the two $[1 \times 1 \times 1]$ and $[k \times k \times k]$ convolutional layers. A squeeze module requires a total parameter of:

$$[C^l \cdot C^l/r] + [C^l/r \cdot C^{l+1}/2] + [(C^l/r) \cdot (k^3 \cdot C^{l+1}/2)] \quad (3)$$

If we set the reduction factor $r = 4$, the kernel size $k = 3$ and double the output feature size $C^{l+1} = 2 \cdot C^l$, our squeeze module attains 7 times reduction in parameter size, as compared to the standard convolutional layer. Replacing convolutional layers in the 3rd, 4th, and 5th feature extraction blocks to squeeze modules reduces the overall model size 4 times.

3) Gaussian Positional Encoding: Positional encoding is a common technique in both computer vision [51] and natural language processing [52], [53] to assist with the training of DNN by providing a positional prior to an input feature.

One unique aspect of ACL tear classification task is that the outer regions of a knee scan do not contain any valuable information for prediction. It is mainly the central part of the scan that contains valuable information for prediction, such as ligaments, bone marrow edema, and anterior tibial translation. Our model incorporates the prior information about the task by adding a 3D Gaussian feature. 3D Gaussian feature softly highlights the central part of a scan by assigning higher values. As shown in figure 3, we concatenate a 3D Gaussian feature to an input feature. We generate 3D Gaussian feature by quantizing 3D Gaussian distribution with $\mu = (\frac{D}{2}, \frac{H}{2}, \frac{W}{2})$ and $\sigma^2 = I_3$.

4) Top-Heavy 13D: To further reduce the model size, we adopt Top-heavy [54] approach for our model design. The top-heavy approach only introduces 3D convolution during the last few feature extraction blocks. As shown in Figure 3, we apply 3D convolution only at the 5th feature extraction block.

A Knee MRI scan has large slice thicknesses (≈ 4 mm). This implies that there is relatively little information to gather across slices than within a slice. Therefore, it is reasonable to adopt a Top-heavy design that tries to gather inter-slice information only when the features are more abstract. Also, a Top-heavy design closely simulates a radiologist's knee examination steps. For each slice, a radiologist first seeks high-level features that indicate the sign of ACL tear. The high-level features are aggregated across neighboring slices only at the final decision-making step.

B. Data

We evaluate our model on two knee MRI datasets: the Chiba and Stanford datasets. Our study was approved by the NUS Institutional Review Board (NUS-IRB).

1) Chiba Dataset: A total of 1177 Knee MRI scans were collected from two institutions in Chiba, Japan, between April 1, 2014, and October 31, 2018. The common indications for the knee MRI examinations in the study included acute, chronic pain and injury, trauma. A 3.0 T MRI (Ingenia CX, Philips Medical Systems) with 16 channel transmit/receive knee coil, and a 1.5 T

MRI (Excelart Vantage, Canon Medical Systems) with 7 channel transmit/receive knee coil was used for Institute H and institute K, respectively (the actual names of the institute were masked following the local data protection guidelines).

2) *Stanford Dataset*: A total of 1370 knee MRI scans were collected from Stanford University Medical Center between January 1, 2001, and December 31, 2012. The common indications for the knee MRI scan in the dataset included acute and chronic pain, follow-up or preoperative evaluation, injury/trauma, and other/not provided. GE scanners (GE Discovery, GE Healthcare, Waukesha, WI) with standard knee MRI coil were used for the extraction. Further detail on the data demographics can be found in the original paper [1].

C. Training

1) *Loss*: Both datasets have a severe class imbalance problem. The Stanford and Chiba datasets have approximately 4 and 2 times more positive cases (ACL tear) than negative cases, respectively. There are various ways in treating the class imbalance problem, such as oversampling of positive samples, undersampling of negative cases, or weighted misclassification penalization. To match with previous works' training approaches, we treat the class imbalance problem with the weighted misclassification penalization. Therefore, our loss function is a weighted Cross-Entropy :

$$L_{CE} = -\mathbb{E}_{\mathbf{x}}[\beta \cdot y \cdot \log(\mathbb{P}(y|\mathbf{x})) + (1 - y) \cdot \log(\mathbb{P}(y|\mathbf{x}))] \quad (4)$$

Where β is the proportion between negative and positive cases (i.e. $\beta = \mathbb{P}_{normal}/\mathbb{P}_{acl}$).

2) *Training Pipeline*: We apply identical data-preprocessing and augmentation schemes to all training instances to prevent possible training biases. For both datasets, we preprocess each scan to have a standardized voxel spacing of $[4, 0.72, 0.72]$, and per-scan intensity mean and variance of $\mu_{\mathbf{x}} = 0$ and $\sigma_{\mathbf{x}} = 1$.

We apply two augmentation techniques: 3D affine transform and random volume cropping. The affine transform performs 3 kinds of image distortions: rotation, translation, and scale. We set the possible range of rotation, translation, and scale to $\pm 15^\circ$, ± 10 pixel, and 1 ± 0.1 , respectively. We randomly crop the affine transformed volume to size $[32 \times 256 \times 256]$.

We optimize all models using Adam optimizer [55]. However, we apply varying learning rates across different training instances. A detailed explanation of how we obtain the learning rates is discussed in the next section. We train all models for 150 epochs with a batch size of 10.

3) *Model Evaluation & Parameter Tuning*: The performance evaluation on the Chiba dataset is executed with 5-fold cross-validation. We apply a grid search method on the first iteration of the 5-fold cross-validation to find the optimal learning rate and apply the same learning rate for the remaining 4 iterations. We attempt 3 learning rates: 1e-3, 1e-4 and 5e-5. The best performing learning rate for our models and MRNet models are found to be 1e-3 and 5e-5, respectively. We use prediction outcomes from the 5 folds to estimate the mean and variance of performance metrics (ROC-AUC, Sensitivity, and Specificity).

The Stanford dataset's performance evaluation is executed on the validation set since the test set is sequestered. An identical approach was taken in other works that use the Stanford dataset [2], [56]. Hence, we find the best learning rate with the train set and test directly on the validation set. The best performing learning rate for our models and MRNet models is found to be 1e-3 and 1e-4, respectively.

IV. RESULTS & DISCUSSION

A. Accuracy & Model Size

Table I compares the diagnostic accuracy and the model size of our proposed models to other ACL detection models. "Our+squeeze" is the proposed model configuration illustrated in Figure 3. The second model, "Our+conv," is constructed by replacing squeeze modules from the "Our+squeeze" model with convolutional modules.

Our best performing model, "Our+conv," outperforms all previous models on the Stanford knee dataset evaluation. "Our+conv" model achieves ROC-AUC of 0.983 ± 0.006 and 0.983 on the Chiba and Stanford knee datasets, respectively. "Our+squeeze" model achieves ROC-AUC of 0.977 ± 0.004 and 0.963 on the Chiba and Stanford datasets, respectively.

"Our+squeeze" model has the smallest model size, requiring merely 43 K parameters. Our model is smaller than MRNets with different backbones: VGGNet, AlexNet, and SqueezeNet by 367, 58, and 17 times, respectively.

From the results, we show that small models are sufficient to achieve state-of-the-arts ACL tear diagnosis performance. We also confirm our hypothesis that ACL tear diagnosis is an easy task for AI models due to the low variability of knee scans and locally confined features.

B. Computational Efficiency

Figure 5 shows the computational efficiency of our models in 4 different aspects: inference Floating point operations per second (FLOPs) utilization, inference & train time per scan (s), and model size (MB).

FLOP(G) measures the number of floating-point calculations (such as addition and multiplication) performed per second when running an application. Our proposed model requires only 7 GFLOPs: 4 times smaller than the original MRNet ("MRNet+AlexNet") and 91 times smaller than "MRNet+VGGNet". We compute the inference & training time in Figure 5 by measuring the average computation time required to iterate over 10 scans (batch size of 10) and divide the measured time by the number of scans.

Our model takes 0.02 and 0.03 seconds to test and train on one scan, respectively. Though our model is much smaller than MRNets, we do not observe significant speed improvements over the MRNets. We speculate that this is due to the multiple sub-blocks in a squeeze module that can not be parallelized, as illustrated in Figure 4.

The last plot measures the size of models in MB. We measure the model size simply from PyTorch's `.pt` extension. The plot

TABLE I

DIAGNOSTIC ACCURACY AND MODEL SIZE. EVALUATION RESULTS OBTAINED ON THE CHIBA AND STANFORD KNEE DATASETS. THE TABLE COMPARES THE DIAGNOSTIC PERFORMANCE OF OUR PROPOSED MODELS TO OTHER ACL TEAR DETECTION MODELS. THE HIGHEST PERFORMANCE FOR EACH EVALUATION METRIC IS HIGHLIGHTED IN BOLD FONT. WE USE 5-FOLD CROSS-VALIDATION TO EVALUATE ON THE CHIBA DATASETS AND TRAIN & TEST SET EVALUATION ON THE STANFORD DATASET. WE OMIT THE EVALUATION OF ELNET [2] AND OTHER WORKS [57], [58] ON THE CHIBA DATASET SINCE THEIR IMPLEMENTATION CODES ARE NOT PUBLICLY AVAILABLE

Model	Chiba			Stanford			# parameter
	Sensitivity±std	Specificity±std	ROC-AUC±std	Sensitivity	Specificity	ROC-AUC	
MRNet+VGGNet	0.943±0.011	0.961±0.008	0.980±0.005	0.944	0.909	0.979	15,715,201
MRNet+AlexNet	0.908±0.026	0.934±0.016	0.973±0.004	0.889	0.894	0.957	2,469,953
MRNet+SqueezeNet	0.932±0.052	0.957±0.016	0.974±0.014	0.778	0.803	0.870	735,937
ELNet [2]	-	-	-	0.923	0.891	0.960	211,314
Irmakci <i>et al.</i> [57]	-	-	-	0.939	0.685	0.955	-
Dunnhoferet <i>al.</i> [58]	-	-	-	0.976	0.815	0.944	-
OUR+Conv	0.930±0.029	0.975 ± 0.013	0.983 ± 0.006	0.981	0.924	0.983	161,921
OUR+Squeeze	0.950 ± 0.023	0.945±0.019	0.977±0.004	0.944	0.901	0.963	42,765

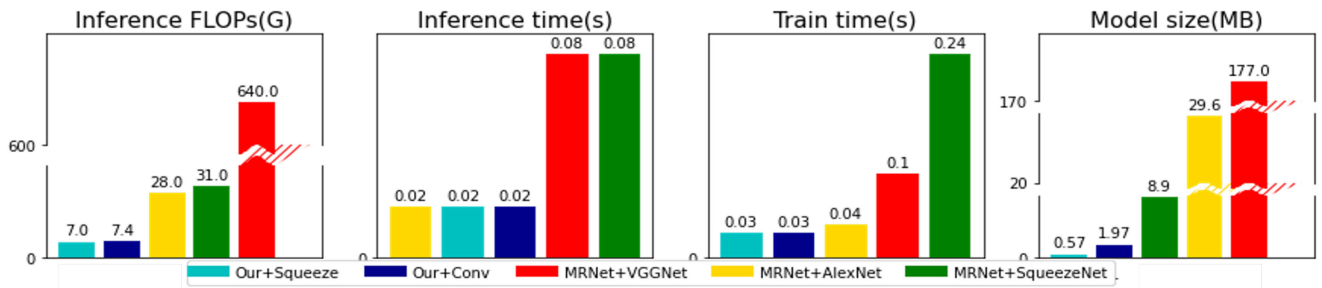


Fig. 5. **Computational cost, speed, and size.** The figure displays the computational efficiency of our models and MRNets in 4 different aspects: Floating point operations per second (FLOPs) utilization, inference & train time per scan (s), and model size (MB).

shows that our model is 15 times smaller than the original MRNet configuration.

We obtain the values using a 3D input with size $[16 \times 256 \times 256]$. We carry out the experiment on a G560 V5 server with a Tesla V100 GPU and an Intel(R) Xeon(R) Gold 6138 CPU. We use PyTorch as our model framework.

C. Visual Interpretation

Figure 6 demonstrates our model’s ability to precisely locate ACL regions from both the Stanford and Chiba knee datasets. We generate the final attention map \mathbf{a} by adding the two attention maps generated from the 3th and 4th feature extraction modules:

$$\mathbf{a}_{d,h,w} = \langle \mathbf{g}_{d,h,w}, \mathbf{z}_{d,h,w}^3 \rangle + \langle \mathbf{g}_{d,h,w}, \mathbf{z}_{d,h,w}^4 \rangle \quad (5)$$

We automatically locate the slice with ACL by finding the depth d which gives the highest attention value (i.e. $\text{argmax}_d(\mathbf{a})$).

Interestingly, the attention maps from the Chiba dataset are noticeably sharper compared to the Stanford dataset. Both MR sequences are water-sensitive sequences to detect edema or hemorrhage from ACL tears. However, the Chiba dataset’s MR sequence is a non-fat suppressed sequence, while that from Stanford is a fat-suppressed sequence. With a non-fat suppressed sequence, the contrast between ACL and surrounding fat is much higher. That possibly helps our model to make a more confident judgment.

TABLE II
EFFECT OF ATTENTION MODULE ON DIAGNOSTIC ACCURACY

Model	Sensitivity±std	Specificity±std	ROC-AUC±std
OUR+ATT	0.926±0.028	0.94±0.036	0.975±0.005
OUR-ATT	0.930 ± 0.029	0.975 ± 0.013	0.983 ± 0.0006

Figure 7 shows a more detailed evaluation of our model’s visualization capability and prediction capability. The percentage is our model’s confidence that the scan has an ACL tear. For the top row scan, our model is very sure, with 92 % confidence, that the image has an ACL tear. For the bottom scan, our model predicts that a tear is less likely with 21%. The attention map does not highlight the torn area (arrows) to help radiologists identify the missing ACL.

D. Ablation Studies

In this section, we assess the effect of attention module and squeeze module on diagnostic accuracy. All experimental results are obtained with 5-fold cross-validation on the Chiba dataset.

Table II shows that adding attention modules not only helps in generating heatmaps but also marginally improves performance. Our model with attention module (“OUR+ATT”) achieves

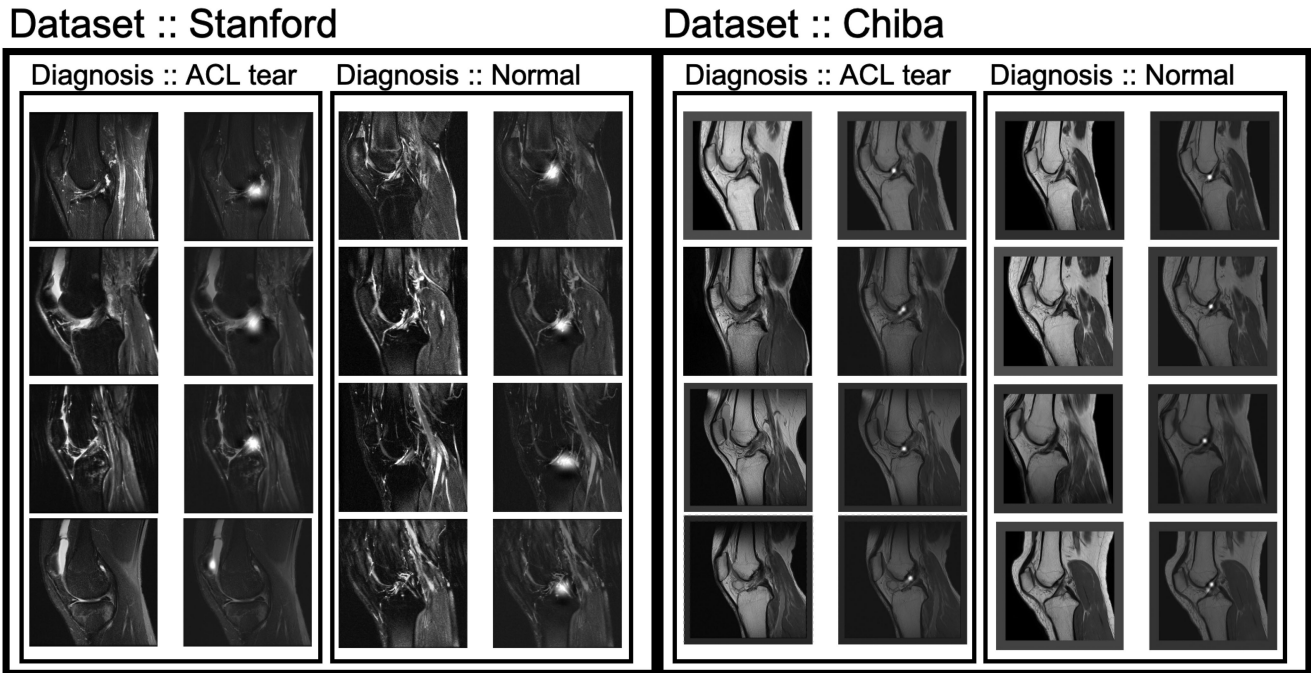


Fig. 6. Heatmaps generated from our attention module. MR sequence from Chiba is a non-fat suppressed sequence, while that from Stanford is a fat-suppressed sequence. The images are unbiasedly sampled from the test set of each dataset. Note the sharp attentions that correctly focus on the ACL region in all cases.

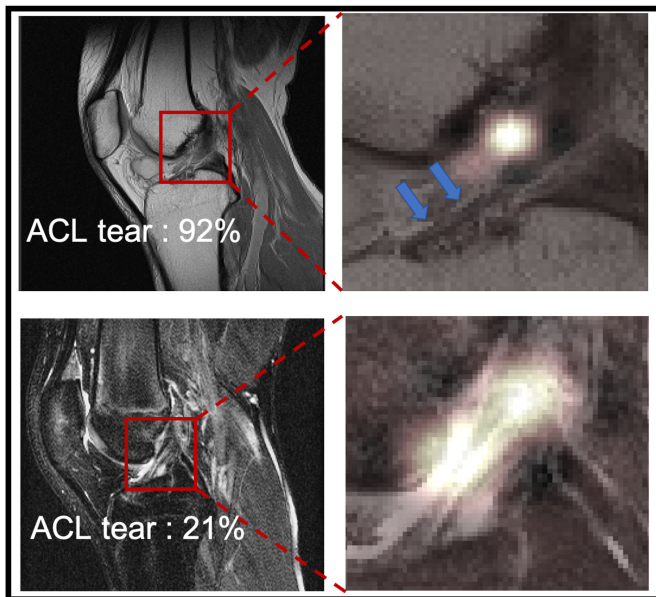


Fig. 7. Automated ACL tear localization. Our attention module automatically selects the MR sequence with an ACL, and within the slice, zooms in precisely to a location where ACL exists. The percentage is our model’s confidence that the scan has an ACL tear. For the top row picture, our model is very sure that the image has an ACL tear. For the bottom picture, our model predicts that a tear is less likely with 21%.

higher performance than the model without attention (“OUR-ATT”) in all metrics. “OUR-ATT” model makes a prediction using the features from the last feature extraction block $g = z^L$ by feeding the feature directly to a dense layer.

TABLE III

EFFECT OF SQUEEZE MODULE WITH VARYING REDUCTION RATE ON DIAGNOSTIC ACCRUACY

Model	Sensitivity \pm std	Specificity \pm std	ROC-AUC \pm std	# parameter
OUR (r = 16)	0.908 \pm 0.084	0.928 \pm 0.025	0.956 \pm 0.035	19K
OUR (r = 8)	0.920 \pm 0.028	0.945 \pm 0.017	0.976 \pm 0.009	25k
OUR (r = 4)	0.950 \pm 0.023	0.945 \pm 0.019	0.977 \pm 0.004	42K
OUR (r = 2)	0.936 \pm 0.024	0.955 \pm 0.020	0.982 \pm 0.007	60K

Table III shows that increasing the squeeze module’s reduction factor (r) can greatly reduce the model size; however, diagnostic performance is marginally sacrificed in return. Nevertheless, we notice that the drop in performance is not drastic compared to the drop in model size. From this observation, we confirm that our squeeze module effectively reduces model size while keeping the performance fairly consistent.

V. LIMITATION & FUTURE WORK

A. Generalization to Other Knee Injuries

There are four major ligaments of the knee: ACL, posterior cruciate ligament (PCL), medial collateral ligament (MCL), and lateral collateral ligament (LCL). In actual clinical settings, in many cases, more than one ligament can be injured. Radiologists are expected to diagnose these ligamentous injuries accurately, and so is AI. We are planning to expand our lightweight algorithm to multiple knee injuries. Our small and fast model will allow us to make this application widely accessible. Especially in busy emergency departments, fast and accurate diagnosis of multiple ligamentous injuries from acute knee trauma will help

orthopedic surgeons plan patients' treatment and surgery in a timely fashion. We will implement our application in emergency departments and prospectively evaluate the usefulness of AI diagnosis of multiple knee ligament injuries after acute traumas such as motor vehicle accidents.

Also, our lightweight AI algorithm is vendor-neutral. It can learn from various MRI vendors quickly and implement any machines easily. We will investigate the accuracy of our model using multiple MRI vendors and different magnetic field machines.

B. Application to Other Medical Imaging Tasks

As aforementioned, our model makes two strong assumptions about a task: 1) the task should rely on a small set of features, and 2) the features should be well confined. These assumptions can be met in many medical imaging tasks other than the ACL injury tasks, such as bone fracture diagnosis and various chest X-ray screening. These tasks commonly have fairly consistent imaging quality (lighting, viewpoint, and scale) and have few class labels to predict.

However, not all medical imaging tasks meet the requirements. Our model is expected to perform poorly on tasks that require many non-local features and have poor imaging quality. Such tasks include mammogram screening and skin lesion analysis. Unlike X-ray or MRI, mammography and dermoscopic image have high inter-image variances. Also, a model should learn to access non-local semantic features such as the distribution of micro-calcifications and shapes of skin lesions.

VI. CONCLUSION

We have demonstrated the possibility to create an interpretable and lightweight AI system for ACL tear diagnosis. Our model can precisely locate ACL and only requires 43 K trainable parameters. Our model's diagnostic accuracy in detecting ACL tear surpasses previously proposed models on the two knee MRI datasets. The findings from our study confirms the importance of having deeper understandings about medical imaging tasks to achieve a model that is interpretable, small and accurate.

REFERENCES

- [1] N. Bien *et al.*, "Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation of MRNet," *PLoS Med.*, vol. 15, no. 11, 2018, Art. no. 1002699, [Online]. Available: <https://www.mdpi.com/1424-8220/19/13/2969>
- [2] C.-H. Tsai, N. Kiryati, E. Konen, I. Eshed, and A. Mayer, "Knee injury detection using mri with efficiently-layered network (ELNET)," 2020, *arXiv:2005.02706*.
- [3] W. H. Ng, J. F. Griffith, E. H. Hung, B. Paunipagar, B. K. Law, and P. S. Yung, "Imaging of the anterior cruciate ligament," *World J. Orthop.*, vol. 2, no. 8, pp. 75–84, 2011. [Online]. Available: <https://doi.org/10.5312/wjo.v2.i8.75>
- [4] P. D. Chang, T. T. Wong, and M. J. Rasiej, "Deep learning for detection of complete anterior cruciate ligament tear," *J. Digit. Imag.*, pp. 1–7, 2019.
- [5] F. Liu *et al.*, "Fully automated diagnosis of anterior cruciate ligament tears on knee mr images by using deep learning," *Radiol. Artif. Intell.*, vol. 1, no. 3, 2019, Art. no. 180091. [Online]. Available: <https://doi.org/10.1148/ryai.2019180091>
- [6] M. A. Ahmad, C. Eckert, and A. Teredesai, "Interpretable machine learning in healthcare," in *Proc. ACM Int. Conf. Bioinf., Comput. Biol., Health Inform.*, 2018, pp. 559–560.
- [7] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.
- [8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.
- [9] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM: Generalized gradient-based visual explanations for deep convolutional networks," 2017, *arXiv:1710.11063*.
- [10] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 6848–6856.
- [11] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and 0.5mb model size," 2016, *arXiv:1602.07360*.
- [12] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
- [13] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, and B. He, "A Survey on federated learning systems: Vision, hype and reality for data privacy and protection," 2019, *arXiv:1907.09693*.
- [14] J. Li, X. Shen, L. Chen, and J. Chen, "Bandwidth slicing to boost federated learning in edge computing," *arXiv*, vol. abs/1911.07615, 2019.
- [15] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Comput. Vision-ECCV 2014*, Springer, 2014, pp. 818–833.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.
- [17] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
- [18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.
- [19] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," 2016, *arXiv:1611.03530*.
- [20] D. Arpit *et al.*, "A closer look at memorization in deep networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2017, pp. 233–242.
- [21] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1580–1589.
- [22] T. B. Brown *et al.*, "Language models are few-shot learners," 2020, *arXiv:2005.14165*.
- [23] Aäron van den Oord *et al.*, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.
- [24] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 191–198.
- [25] A. Vellido, "The importance of interpretability and visualization in machine learning for applications in medicine and health care," *Neural Comput. Applications*, pp. 1–15, 2019.
- [26] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Computer Vis.*, Springer, 2014, pp. 818–833.
- [27] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?" Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discov. Data Mining*, 2016, pp. 1135–1144.
- [28] J. Irvin *et al.*, "CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, 2019, pp. 590–597.
- [29] H. Alshazly, C. Linse, E. Barth, and T. Martinetz, "Explainable COVID-19 detection using chest CT scans and deep learning," *Sensors*, vol. 21, no. 2, 2021, Art. no. 455.
- [30] T. Zebin and S. Rezvy, "COVID-19 detection and disease progression visualization: Deep learning on chest X-rays for classification and coarse localization," *Appl. Intell.*, vol. 51, no. 2, pp. 1010–1021, 2021.
- [31] P. Rajpurkar *et al.*, "MURA: Large dataset for abnormality detection in musculoskeletal radiographs," 2017, *arXiv:1712.06957*.
- [32] C.-T. Cheng *et al.*, "Application of a deep learning algorithm for detection and visualization of hip fractures on plain pelvic radiographs," *Eur. radiol.*, vol. 29, no. 10, pp. 5469–5477, 2019.

- [33] R. Lindsey *et al.*, "Deep neural network improves fracture detection by clinicians," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 45, pp. 11 591–11 596, 2018.
- [34] V. Golkov *et al.*, "q-space deep learning for Alzheimer's disease diagnosis: Global prediction and weakly-supervised localization," *Proc. 27th Annu. Meeting ISMRM*, Paris, France, vol. 1580, 2018.
- [35] T. Iizuka, M. Fukasawa, and M. Kameyama, "Deep-learning-based imaging-classification identified cingulate island sign in dementia with lewy bodies," *Sci. Rep.*, vol. 9, no. 1, pp. 1–9, 2019.
- [36] I. P. De Sousa, M. M. B. R. Vellasco, and E. C. DaSilva, "Local interpretable model-agnostic explanations for classification of lymph node metastases," *Sensors* (Basel, Switzerland), vol. 19, no. 13, 2019, [Online]. Available: <https://www.mdpi.com/1424-8220/19/13/2969>
- [37] F. Wang *et al.*, "Residual attention network for image classification," in *Proc. IEEE Conf. Computer Vis. Pattern Recognit.*, 2017, pp. 3156–3164.
- [38] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Computer Vis.*, 2018, pp. 3–19.
- [39] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Computer Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
- [40] P. H. Seo, Z. Lin, S. Cohen, X. Shen, and B. Han, "Progressive attention networks for visual attribute prediction," 2016, *arXiv:1606.02393*.
- [41] S. Jetley, N. A. Lord, N. Lee, and P. H. S. Torr, "Learn to pay attention," in *Proc. 6th Int. Conf. Learn. Representations*, ICLR 2018, Vancouver, BC, Canada, Apr. 30 - May 3, 2018, Conf. Track Proc. OpenReview.net, 2018. [Online]. Available: <https://openreview.net/forum?id=HyzbhfWRW>
- [42] O. Oktay *et al.*, "Attention U-Net: Learning where to look for the pancreas," 2018, *arXiv:1804.03999*.
- [43] J. Schlemper *et al.*, "Attention-gated networks for improving ultrasound scan plane detection," *Medical Image Comput. Comput.-Assisted Intervention: MICCAI—Int. Conf. Medical Image Comput. Comput.-Assisted Intervention*, Apr. 2018, submitted to MIDL2018 OpenReview: <https://openreview.net/forum?id=BJtn7-3sM>
- [44] Y. Hu *et al.*, "AIDAN: An attention-guided dual-path network for pediatric echocardiography segmentation," *IEEE Access*, vol. 8, pp. 29 176–29 187, 2020.
- [45] J. Li, W. Li, A. Gertych, B. S. Knudsen, W. Speier, and C. W. Arnold, "An attention-based multi-resolution model for prostate whole slide image classification and localization," 2019, *arXiv:1905.13208*.
- [46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.
- [47] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2015, pp. 448–456.
- [48] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*.
- [49] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, "Speeding-up convolutional neural networks using fine-tuned cp-decomposition," 2014, *arXiv:1412.6553*.
- [50] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Computer Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
- [51] R. Liu *et al.*, "An intriguing failing of convolutional neural networks and the coordconv solution," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9605–9616.
- [52] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North American Chapter Association Comput. Linguistics: Human Lang. Tech., Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: <https://www.aclweb.org/anthology/N19-1423>
- [53] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [54] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 305–321.
- [55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [56] M. J. Awan, M. S. M. Rahim, N. Salim, M. A. Mohammed, B. Garcia-Zapirain, and K. H. Abdulkareem, "Efficient detection of knee anterior cruciate ligament from magnetic resonance imaging using deep learning approach," *Diagnostics*, vol. 11, no. 1, 2021.
- [57] I. Irmakci, S. M. Anwar, D. A. Torigian, and U. Bagci, "Deep learning for musculoskeletal image analysis," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, IEEE, 2019, pp. 1481–1485.
- [58] M. Dunnhofer, N. Martinel, and C. Micheloni, "Improving MRI-based knee disorder diagnosis with pyramidal feature details," in *Proc. Fourth Conf. Medical Imaging Deep Learn. (MIDL)*, 2021.