1. Ultrasound Data Preprocessing

1.1 Image Cropping

To crop the important area out of the unrelated regions in ROI of the image, threshold segmentation and morphology methods were employed. Notably, the pixel values corresponding to the bone structures consistently exceeded a threshold of 0. Thus, we applied this threshold to effectively segment the bone structures. To address narrow gaps and refine the contours, morphological closing and opening operations were applied. We obtained the area of importance by employing a series of close-open-close operations. Subsequently, a rectangular region measuring 700 x 320 pixels was cropped. This size effectively covers all significant regions while minimizing unrelated areas.

1.2 Normalization

The variability in pixel value distribution could introduce classification errors. In order to standardize the pixel value distribution, we applied a normalization procedure. This involved calculating the mean value of all pixel values within the image sample, subtracting this mean value from each individual pixel value, and subsequently dividing the result by the maximum value obtained from the subtraction.

2. Model development

2.1 The deep learning models

Motivated by ResNet18 model, we used Residual blocks to construct a lightweight model to complete the classification task. The core of ResNet was the pivotal Residual block. Prior to the conceptualization of ResNet, the challenge of vanishing/exploding gradients posed significant obstacles as layer stacks grew larger. ResNet ingeniously addresses this issue by utilizing Residual blocks. Instead of making each few stacked layers directly fit a desired underlying mapping, the Residual block explicitly let these layers fit a residual mapping. Assuming the fundamental mapping as H(x), the Residual block guides the accumulation of nonlinear layers to model another mapping denoted as F(x): = H(x)−x. This mechanism employs "short-cut connections" to effectuate the F(x)+x relationship, thus giving rise to the Residual block structure. Comprising two convolutional layers for feature extraction, the Residual block integrates a short-cut connection that links the input and output. In this study, we leverage three Residual blocks to establish the network architecture. Following the feature extraction by the Residual blocks, a fully-connected layer was employed to integrate all extracted features and convert them into a two-dimensional vector. The layers of our ResNet are illustrated in the table 1.

2.2 Cross validation

As our dataset size was relatively modest, we employed cross-validation as a crucial methodology. The proportions of patients in the gout group and the control group were equivalent to prevent the concurrent use of images from the same individual for both training and testing. We choose one fold as testing set, the other four folds were used as training set. Each patient has several images, all these images from patients in training set were mixed up to train the model. The images from patients in testing set were mixed up to test the model. Then we choose the next fold as testing set, and repeat the above operation. Analogously, we repeat the operation five times and get the average results.

2.3 Data augmentation

Data augmentation was used to expand the dataset. We applied translations to the images in the training set to generate additional data. Consequently, the size of the training set was tripled, enhancing the model's ability to better adapt to the dataset with increased data.

2.4 Training

This experiment was built on x86_ 64 processor Linux 4.4.0 operating system, experimental environment using the Pytorch 1.7.1 framework, programming language using Python 3.8, visual studio 1.75.1 as editor, CPU using Intel (R) Xeon (R) CPU E5-2620 v4 @ 2.10GHz, GPU using NVIDIA GeForce RTX 3090, running memory of 24GB, all programs implemented through Pytorch' s opensource framework.

The full-connection layer output was a two-dimensional vector. Softmax was used to convert it into probability. The probability of two classes was in range of [0,1], and the sum was 1.

$$y_1 = \frac{e^{x_1}}{e^{x_1}+e^{x_2}} \quad (A.1)$$

$$y_2 = \frac{e^{x_2}}{e^{x_1}+e^{x_2}} \quad (A.2)$$

We use Cross entropy Loss function as loss function (17). It use the probability of two class to calculate the cross entropy. The little the cross entropy was, the better the model fits the dataset. We use the Adam optimizer to train the model so that could reduce the loss.

$$L = \frac{1}{N}\sum_i -[y_i \log(p_i) + (1-y_i)\log(1-p_i)] \quad (A.3)$$

The learning rate was 1e-5, batch size was 32 and the training epoch number was 30. The images were shuffled after data processing, and then fed into the model.

3. Heat map generation

During the testing process, the Pytorch framework was applied to extract the last convolutional layer of the data model and obtain the feature map. By using the Gradient-weighted Class Activation Mapping (grad-CAMs) function, the trained data model and the extracted feature maps were inputted to obtain the grad-CAMs map. Overlaying these maps onto the original images facilitat-

ed the creation of thermal maps for auxiliary diagnostic purpose. The grad-CAMs function was a function code of the Pytorch framework, which could use the feature map of the last layer of the model and the category determined by the model to calculate and obtain the thermal map of the model's focus.

Table S I. The layers of our ResNet

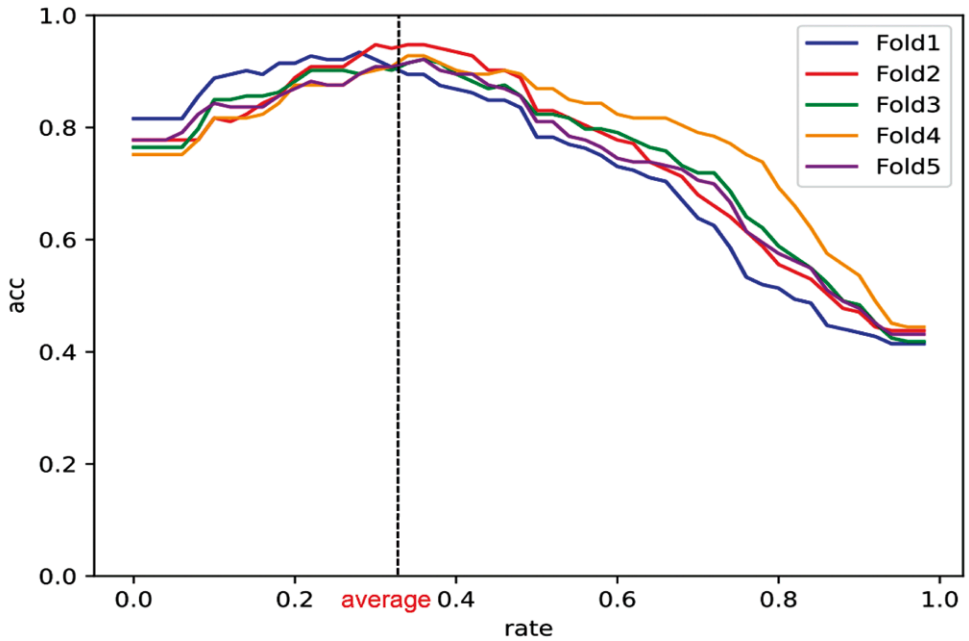
Layer Name		Conv Kernel	Stride	Output (feature map size,channels)
Conv1		3×3	2	350×160,64
Maxpooling		5×5	2	174×79,64
Residual block1	conv	3×3	2	87×40,128
	conv	3×3	1	87×40,128
Residual block2	conv	3×3	2	44×20,256
	conv	3×3	1	44×20,256
Residual block3	conv	3×3	2	22×10,512
	conv	3×3	1	22×10,512
Average pooling		22×10	1	1×1,512
Full connected layer				2
Softmax				

Conv, Convolution.

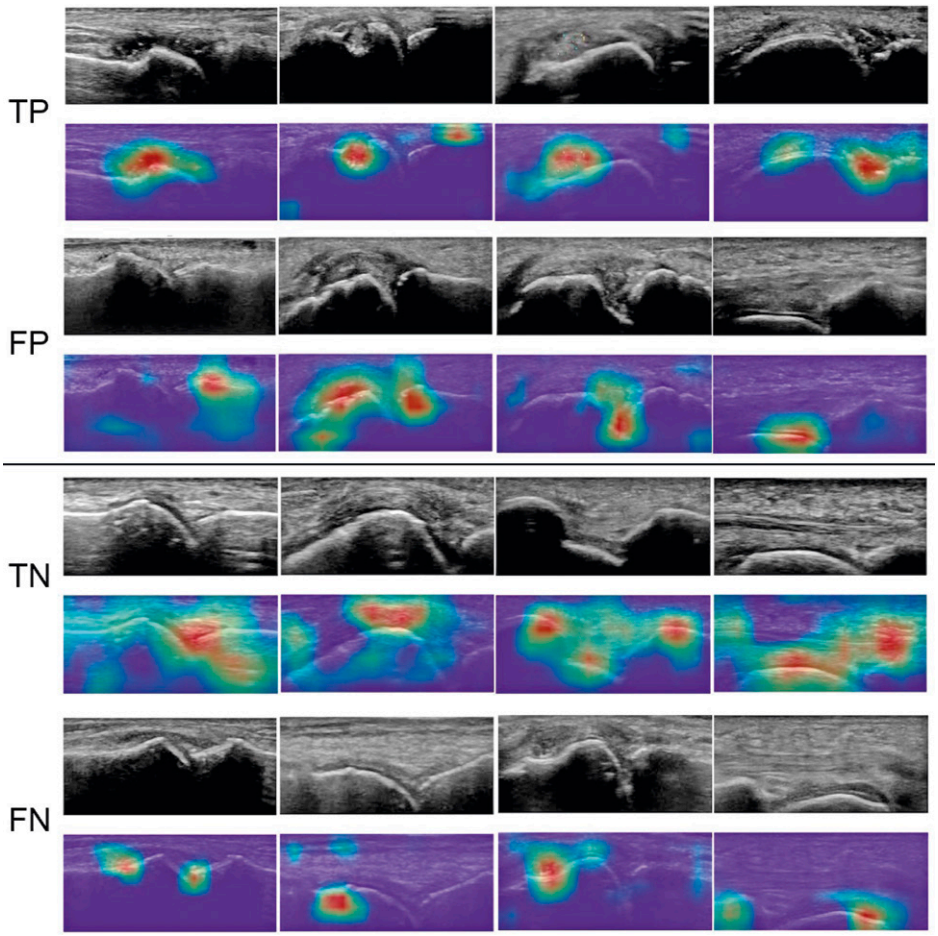
4. The calculation formulas:

Accuracy = (TP+TN) / (TP+TN+FP+FN)
Sensitivity = TP / (TP+FN)
Specificity = TN / (TN+FP)

TP, Ture Positive; TN, Ture Negative; FP, False Positive; FN, False Negative.



Supplementary figure 1. Five-fold Threshold Curves of the Training Set
rate: Proportion of Abnormal Images; average: Average Threshold; acc, accuracy.



Supplementary figure 2. Typical Images of Correct and Incorrect Classification (Ultrasound images and Heatmaps)
TP: Typical features of GA (aggregates, bone erosion, tophus, and DCs); FP: Normal areas that are easily mistaken for abnormalities (irregular bone texture, relatively full medial collateral ligament, anisotropy artifact, and "interface effect"); TN: Normal images of the three parts of the first MTPJ; FN: Easily overlooked abnormal lesions (scattered aggregates, slight synovial thickening, and discontinuous DCs). GA, Gouty Arthritis; DCs, Double-Contour Sign; MTPJ, Metatarsophalangeal Joint.