**LEBANESE AMERICAN UNIVESRITY**

**COMPUTER SCIENCE AND MATH DEPARTMENT, BYBLOS**

**CSC615 MACHINE LEARNING**

**FINAL PROJECT  
  
  
Ghadi El Hasbani, LAU I.D.: 201903122**

**Danielle Antoun Azar, PhD**

**Abstract**

Kidney diseases in general and fibrosis specifically are of world-wide importance. Medical ultrasound (US) provides a non-invasive means to identify fibrotic stages in a clinical setting. Deep learning methods try to classify disease stages in an attempt to produce robust and accurate classifications. With sparse data, the use of transfer and ensemble learning could prove useful in improving performance and reducing variance. In this study, the proposed method combines models of various depths, VGG16, ResNet50, InceptionResNetV2, and DenseNet121, to classify fibrotic stages of renal US images from 31 mice. Model performance proved unsatisfactory patient-wise indicating possible overfitting. Although sample-wise performance could also be improved, a useful clinical trend is remarked whereby sham and mild groups show a low false positive rate as compared to a low false negative rate for the severe group. Further work should consider different submodel architecture, possible shallower ones, combined with different training procedures to alleviate overfitting and boost performance and patient-wise generalizability.

**Introduction**

Kidneys are crucial to homeostasis maintenance through their participation in regulation and excretion (Torres et al., 2018). Around 850 million people world-wide suffer from kidney-related diseases as of 2019 (Biradar et al., 2022). Renal fibrosis plays a crucial role in kidney disease progression. For example, in chronic kidney disease (CKD), fibrosis is found to differing degrees across disease stages. CKD is characterized by progressive scaring, loss of renal cells, and increased extracellular matrix production leading to fibrosis affecting the whole kidney. Ideal markers for assessing fibrosis are only found using biopsy which is an invasive method (Peride et al., 2016).

Medical Ultrasound (US) imaging works by emitting a pulse that is reflected at the boundary of different tissue types depending on the difference in their acoustic impedance (Adamo et al., 2013). Moreover, US is capable of capturing information relating to organs structure and movement as well as blood flow. It is a non-invasive technique commonly used for clinical diagnoses involving numerous organs such as the breast, liver, kidney, uterus, and thyroid. This has made US images widely available as compared to other imaging techniques and a common tool for the creation of models predictive of disease in order to assist professionals in the clinical setting. Nevertheless, US imaging, especially hand-held, generates noisy internal snapshots of the body with no clear specification of where each organ is located. Moreover, high-end ultrasound equipment is not always accessible such as in community, rural, and tele- medicine (Zhou et al., 2019). This is also especially true in data recorded from research with animals, especially small-sized animals like mice. Finally, image quality is influenced by staff expertise, and diagnoses is influenced by the doctor’s personal experience (Meng et al., 2017). The human eye is limited in its ability to produce robust and accurate diagnoses. Therefore, the use of US imaging for predictive models is an ongoing area of research presenting various challenges in the field of computer vision.

Research has focused on the 3 main challenges in US-based diagnoses: quality enhancement, segmentation, and classification. Quality enhancement is focused on the reduction of speckle noise using methods such as wavelet decomposition (Adamo et al., 2013) as well as contrast enhancement which are two main quality concerns in kidney US and US in general (Kaur & Singh, 2022). Some methods have also resorted to image restoration using generative methods (Zhou et al., 2019). Computer-aided segmentation consists of identifying meaningful regions in the US image to characterize organs and even tissues to avoid time-intensive and variable manual segmentation (Torres et al., 2018).

Classification methods aim for computer-aided diagnostics that identify between US images of diseased and non-diseased patients. Some methods also aim to distinguish between different types and stages of the same disease (see Chen et al., 2020 for an example of the latter). Although some methods still use classical machine learning, they often employ feature extraction methods beforehand. For example, Biradar et al. (2022) extracted shape, wavelet, Haralick, tamura, and Histogram of Oriented Gradient features before using k-Nearest Neighbors (k-NN), fuzzy k-NN, and support vector machine (SVM) for chronic kidney disease (CKD) classification. Nevertheless, since high-dimensionality and data availability are both issues faced when processing US images, numerous methods have resorted to convolutional neural (CNNs) and long-short-term-memory (LSTM) networks for their ability to handle such complexity, especially in the feature extraction step. To combat data scarcity, transfer learning has been employed through the repurposing of models trained for general object recognition tasks. For example, Meng et al. (2017) employ VGGNet (Simonyan & Zisserman, 2014) trained on ILSVRC (Olga et al., 2014) benchmark data for feature extraction followed by a 3-layer deep fully connected classification component with dropout for US liver fibrosis classification. Misra et al. (2021) propose an ensemble transfer learning (TL) approach combining B-mode breast US (B-US) and strain elastography breast US (SE-US) images to discriminate between benign and malignant breast tumors. The ensemble consisted of AlexNet (Krizhevsky et al., 2017) and ResNet-18 (He et al., 2016) trained on ImageNet (Deng et al., 2009). The models were chosen for their good performance and shallow structures for computational efficiency as well as ResNet’s ability to combat vanishing gradients with skip connections. The classification layer is then dropped from each model and the output of both networks are concatenated. The concatenated output is finally passed to a softmax classification layer. Nevertheless, concatenating submodel representations causes the size of the concatenated layer to increase dramatically with the addition of every transferred model to the ensemble.

In this study, a method combining ensemble and transfer learning is used to classify US images of mouse kidneys into kidney fibrosis stages being sham, mild, and severe. This method is then compared to a variant of an existing method, FCNet (Meng et al., 2017), as well as the individual performance of submodels considered.

**Materials & Methods**

*Data*

The dataset consists of images from 31 mice (8 sham, 9 stage 1 fibrosis, 14 stage 2 fibrosis). The images were manually cropped to only include the kidney. Height and width of each image were downsized to the minimum value in the dataset for each dimension for a resulting image size of 158x275x3 (HxWxC).  In mice, it is possible to induce kidney fibrosis by causing something called Ischemia Reperfusion Injury (IRI). As shown in the schematic below, IRI is induced by surgically exposing the kidney, clamping the renal artery for a precise period of time, releasing the clamp and closing up the abdomen of the mouse. Depending on the duration of the clamp, the degree of IRI (and fibrosis that develops due to this injury) increases the longer the renal artery is blocked. In this experiment, two different levels of IRI were induced: Mild IRI achieved after a 22 min clamp and Severe IRI achieved after a 45 min clamp. The sham group includes mice that have undergone every step of the surgery except vessel clamping to mimic surgery conditions of the other groups. Each mouse was imaged using the VevoLAZR imaging system using a 21 MHz linear array transducer with 256 elements. US/CEUS/PA was performed at various time points. At every imaging time point, 101 2D B-mode frames were acquired by 3D scanning over the entire kidney volume at 100 micrometer step sizes. As the kidney volume might not show up in every frame, it's probably safest to work with frames 30-90. This will of course depend on the imaging orientation on imaging session. Example images from each group are shown in Figure 1. Since color in US images contains information on blood flow, and kidney fibrosis is a circulatory disease, color channels are maintained throughout the analysis.

One-third of the mice in the dataset were reserved for testing (22 training, 9 testing). All training was done using a batch size of 3 images. All code pertaining to this study can be found publicly on GitHub (https://github.com/GhadiElHasbani/CSC615-MachineLearning).

A picture containing text

Description automatically generated

Figure 1 Renal US images of a mouse from each stage: Sham, Mild, Severe from left to right in that order

*Transfer Learning*

Transfer learning is a popular method in deep learning that takes advantage of pre-trained models on large-scale benchmark datasets or any large data on a specific task A and extends this model on a different, most of the times similar, task B. Transfer learning is usually used when data for the task at hand is limited or sparse, and training with the data at hand would not yield satisfactory performance. This is usually because the task at hand is complex. The pre-trained model can then be transferred as is or fine-tuned with the limited data at hand that corresponds to the given task B. Transfer learning is particularly popular in image processing. The availability of large object detection datasets such as ImageNet (Deng et al., 2009) and state-of-the-art object recognition models such as GoogleNet (Szegedy et al., 2015), ResNet (He et al., 2016), VGG-family (Simonyan & Zisserman, 2014) and others enable an opportunity to compensate for sparse image datasets such as in medical imaging. Moreover, the use of deep learning models over static feature extraction methods makes the feature extraction dynamic.

In this study, models of various depths and architectures were chosen for diversity. Deeper models were trained slightly longer than shallower models.

ResNet50 (He et al., 2016) resorts to double and triple-layer bypasses to connect layers at nonconsecutive depths to reduce the risk of the gradient vanishing problem. The layers include 50 layers with skip connections every 3 layers and bottleneck convolutional layers of kernel size 1x1 as well as both max and average pooling. The last 20 layers were fine-tuned with a learning rate of 0.00001. The classification layer was trained using the Adamax optimizer (Kingma & Ba, 2014), a variant of the Adam optimizer based on the infinity norm, and a learning rate of 0.005 for 3 epochs.

VGG16 (Simonyan & Zisserman, 2014) is composed of 13 convolutional layers divided into 2 blocks of 2 and 3 blocks of 3 in that order. Each block is followed by a max pooling layer. The output of the last block is passed to 3 fully connected layers. All activations are ReLU. The last 8 layers were fine-tuned with a learning rate of 0.0001. The classification layer was trained using the NAdam optimizer and a learning rate of 0.001. Training lasted for 3 epochs. The NAdam optimizer (Dozat, 2016) implements Nesterov momentum to the Adam optimizer (Kingma & Ba, 2014) to help combat overfitting by escaping local, or global, minima.

InceptionResNetV2 (Szegedy et al., 2017) is 164-layer network composed of inception modules which have convolutional layers process the output of the previous layer in parallel. This alleviates the depth of the network by having convolutional layers in parallel instead of sequential (Sharma & Guleria, 2022). Connections are also introduced to skip inception blocks. Sequential convolutional layers are also found between these blocks. The model also incorporates both max and average pooling and uses ReLU activation. The last 30 layers were fine-tuned with a learning rate of 0.0001. The classification layer was trained using the NAdam optimizer and a learning rate of 0.005. Training lasted 10 epochs.

DenseNet121 (Huang et al., 2017) is composed of DenseBlocks which maintain the same representation size within. Transition blocks are found between DenseBlocks to reduce representation size. Inside a DenseBlock, each layer is connected to each subsequent layer. DenseNet121 also uses both average and max pooling along with ReLU activations. DenseNet121 uses a similar strategy to ResNet50 whereby it connects each layer with all subsequent layers to combat gradient vanishing (Huang et al., 2017). The last 5 layers were fine-tuned with a learning rate of 0.00001. The classification layer was trained using the NAdam optimizer and a learning rate of 0.005. Training was 15 epochs long.

All training settings were determined using trial and error. All fine-tuning was done using Stochastic Gradient Descent (SGD) optimization. The 1000-unit softmax top layer of all models was dropped prior to any training and a classification layers consisting of a 3-unit softmax layer was added.

All experiments were run on Google Colab GPU using Python’s TensorFlow (Abadi et al., 2016) and Keras (Chollet, F., & others., 2015) libraries.

*Ensemble Learning*

Ensemble Learning (EL) is a useful method that combats model variance. Variance is usually increased with depth of the network and small sample sizes. In the case of ultrasound (US) image processing methods, problems are usually complex and data availability is sparse. Therefore, ensemble methods are used to gain different perspectives on the problem at hand using diverse submodels with a single, integrated output. Ensemble pruning methods can also be useful to select an optimal subset of models to boost prediction.

In this study, all of ResNet50 (He et al., 2016), VGG16 (Simonyan & Zisserman, 2014), InceptionResNetV2 (Szegedy et al., 2017), and DenseNet121 (Huang et al., 2017) were combined in an ensemble model using majority vote.

A variant of FCNet was also constructed by dropping the softmax layer of every submodel followed by a concatenation of the output of each model as proposed in Misra et al. (2021). The concatenated output is then processed by a 3-layer deep fully-connected network. Each layer had an L2 regularization of 0.2 and a dropout of 0.1. All 3 layers shared the same PReLU activation function parameters corresponding to each unit and hence had the same number of units (552). The final layer is a 3-unit softmax classification layer. FCNet was trained for 10 epochs using the NAdam optimizer and a learning rate of 0.0001. For FCNet’s, the submodels were only trained on two-thirds of the training set whereas FCNet itself was trained on the remaining third.

*Activation function*

Neural networks can be built with varying architectures and hyperparameters. Each unit has a specified activation function to relay weighted output of the previous layer while introducing non-linearities. Therefore, the choice of function is an important consideration and can affect network performance. Non-linearities are also an advantage that these functions present to enable complex feature extraction. Activation functions can be generally grouped into saturated and unsaturated functions. Saturated functions are those like the sigmoid function which is only sensitive to mid-range values while saturating close to 0 and 1. This means that it is possible for sigmoid units to cause a 0 gradient with increasing number of layers. The consequence would be loss of information (Tan & Lim, 2019). This is because at each layer, the gradient is the product of the gradient flow and the local gradient, whereby the former is influenced by deeper layers and the latter is influenced by the choice of activation. Inefficiently small gradients are then more likely with increasing depth, which increases the likelihood of a small value being propagated, and saturated activation functions. Having all local gradients between 0 and 1, such as is the case with saturated functions, will result in a vanishing gradient (Kong & Takatsuka, 2017).

Rectified linear units (ReLUs) were introduced to mitigate the issue of saturated activation functions. Unlike the sigmoid function, ReLU is sensitive to all positive inputs which made it a popular choice in deep and wide networks since it can overcome the vanishing gradient by being unsaturated. On the other hand, since ReLU’s output can be any positive value, it is prone to cause an exploding gradient where drastic weight updates are performed. It is therefore recommended to have pretrained or properly initialized weights. Moreover, an output of 0 combined with a negatively biased input will result in a 0 gradient and what is dubbed as the dying ReLU problem as neurons deactivate without possibility of reactivation (Tan & Lim, 2019). Leaky ReLU (LReLU) is a modified version of ReLU with a, usually, positive configurable slope before 0 that allows for negative output (Maas et al., 2013). The slope value is trained in parametric ReLU (PReLU; He et al., 2015). For these reasons, PReLU is the activation function of choice in this analysis.

**Results & Discussion**

In this study, all results are reported sample-wise and patient-wise on the testing set over n=30 iterations each having differently seeded weight initializations. For patient-wise scores, predictions considered are the majority vote for each patient based on sample-wise predictions of that patient. Metrics reported are multiclass accuracy (mAC) and one-versus-all accuracy (AC), f1-Score (F1), specificity, recall (RE), precision (PR), and area under the receiver-operational curve (AUC) for each class. All these metrics are reported for the ensemble. The multiclass and one-versus-all accuracy is additionally reported for the submodels sample-wise and patient-wise in addition to the sample-wise categorical cross-entropy loss.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| (%) | **Accuracy** | **AUC** | **Precision** | **Recall** | **Specificity** | **F1-Score** |
| Sham vs All | 78.72 ±01.42 | 53.69 ±03.11 | 73.95 ±23.88 | 8.64 ±06.89 | 98.74 ±01.69 | 17.33 ±08.84 |
| Mild vs All | 84.48 ±03.92 | 77.40 ±05.95 | 95.74 ±05.10 | 56.16 ±12.35 | 98.64 ±01.92 | 69.94 ±09.94 |
| Severe vs All | 65.56 ±04.60 | 68.84 ±04.07 | 56.71 ±03.70 | 98.41 ±03.11 | 39.27 ±09.19 | 71.85 ±02.66 |

Table Sample-wise one-versus-all metrics of majority vote ensemble (% mean ±standard deviation (sd)) over n=30 iterations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| (%) | **Accuracy** | **AUC** | **Precision** | **Recall** | **Specificity** | **F1-Score** |
| Sham vs All | 77.41 ±02.03 | 49.76 ±01.30 | 00.00 ±00.00 | 00.00 ±00.00 | 99.52 ±02.61 | - |
| Mild vs All | 68.52 ±05.12 | 60.28 ±06.45 | 55.00 ±18.65 | 35.56 ±12.17 | 85.00 ±05.09 | 43.79 ±08.49 |
| Severe vs All | 46.67 ±06.12 | 49.75 ±05.92 | 44.44 ±04.16 | 77.50 ±07.63 | 22.00 ±09.61 | 56.37 ±04.42 |

Table Patient-wise one-versus-all metrics of majority vote ensemble (% mean ±standard deviation (sd)) over n=30 iterations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Accuracy(%) | **Majority Vote** | **FCNet** | **ResNet50** | **DenseNet121** | **InceptionResNetV2** | **VGG16** |
| Patient-wise | 46.30 ±05.12 | 41.11 ±09.30 | 48.52 ±09.45 | 40.00 ±11.15 | 43.70 ±09.20 | 45.56 ±05.34 |
| Sample-wise | 64.38 ±04.49 | 60.81 ±08.35 | 62.96 ±05.22 | 58.96 ±08.66 | 60.27 ±06.15 | 66.09 ±06.17 |

Table Patient-wise and sample-wise multiclass accuracy (mAC) of all candidate models (% mean ±standard deviation (sd)) over n=30 iterations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Loss | **FCNet** | **ResNet50** | **DenseNet121** | **InceptionResNetV2** | **VGG16** |
| Sample-wise | 17.47 ±01.35 | 09.10 ±01.94 | 74.95 ±23.93 | 22.91 ±06.69 | 01.10 ±00.35 |

Table Sample-wise categorical cross-entropy loss of FCNet and submodels (mean ±standard deviation (sd)) over n=30 iterations

In regards to sample-wise mAC, the majority vote ensemble (64.38% ±04.49) outperforms all candidate models except VGG16 (66.09% ±06.17). Although, if variance is taken into account, both models seem to perform comparably. This raises the question of whether VGG16’s performance alone is sufficient, and adding more models does not improve performance in this case. FCNet (60.81% ±08.35) performs comparably to (60.27% ±06.15) which outperform DenseNet121 (58.96% ±08.66) and which are in turn outperformed by ResNet50 (62.96% ±05.22). Performs seems to decline with an increase in depth of the considered model although this is not the case for DenseNet121 as compared to InceptionResNetV2. Nevertheless, this supports the idea that deeper models are less suitable for small datasets as compared to shallower ones. This indicates that shallower candidates could have improved performance. The sample-wise categorical cross-entropy loss shows a similar trend with VGG16 having the lowest score (01.10 ±00.35). This supports the previous observation on model depth. FCNet (17.47 ±01.35) achieves a loss slightly lower than that of InceptionResNetV2 (22.91 ±06.69) indicating that it does not perform well in combining submodels.

For patient-wise mAC, the top model is ResNet50 (48.52% ±09.45) with majority vote ensemble being second best (46.30 ±05.12). Although, all models score dramatically worse on this metric as compared to sample-wise mAC. This indicates that models often fail to generalize to images of the same mouse/patient indicating possible overfitting.

In regards to one-versus-all metrics, the proposed methods performs exceptionally well for sham and mild groups as indicated by both sample- (sham, 98.74% ±01.69; mild, 98.64% ±01.92) and patient-wise (sham, 98.74% ±01.69; mild, 85.00% ±05.09) specificity. Nevertheless, sample- (39.27% ±09.19) and patient-wise (22.00% ±09.61) specificity is low for the severe group. On the other hand, recall indicates best performance for the severe group sample- (98.41% ±03.11) and patient-wise (77.50% ±07.63). This indicates a low false negative rate for the severe group but a low false positive rate for the sham and mild groups. This is a trend that could prove useful in the clinical setting with some improvement. AUC scores are only relatively acceptable for sample-wise mild (77.40% ±05.95) and severe (68.84% ±04.07) groups as well as the patient-wise mild group (60.28 ±06.45). Most importantly, patient-wise precision and recall for the sham group are both 0 indicating that none of the true positives have generalized in this group. Sample-wise recall is also already low sample-wise for this group (8.64 ±06.89). Moreover, the high sample-wise precision of the mild group (95.74 ±05.10) indicates a low false positive rate. This group also has the highest sample-wise accuracy (84.48% ±03.92) as compared to the sham group with the highest patient-wise accuracy (77.41% ±02.03). In all, it seems the proposed method does not generalize well patient-wise and a different training procedure and submodel architectures should be considered to overcome overfitting. Nevertheless, sample-wise measures indicate a meaningful trend in the model’s ability to distinguish the three groups. Specifically, the model exhibits low sample-wise false negative rates for the severe group as compared to a low false positive rate in the remaining groups. Further improvement could be achieved for true positives in all cases. Finally, the ideal trend in this case would be if the severe and mild groups both had low false negative rates as compared to a low false positive rate for the sham group combined with high true negative rates in all groups.

**Limitations & Future Work**

The main limitation of this study is a lack of thorough preprocessing. Ideally, image restoration would be employed along with possible segmentation to enhance the ensemble input. Moreover, image data augmentation could be employed, including methods specific to US imaging such as those described in Lee et al. (2021). Feature extraction methods could also be used to supplement CNN transferred feature extractors. The combined output could then be further processed with different subsequent learners. A good candidate to explore as a meta-learner could be a ranking model which aims to order the samples instead of classify them. The ordered severity scores could then be used to generate categorical output. With time-dependent data which captures the same mice at different fibrotic stages could be useful for real-time severity score quantification and disease classification using Disease Severity Score Learning (DSSL) as proposed in Dyagilev & Saria (2016). Finally, overfitting is a big concern with small datasets, and shallower models could contribute to increasing classification performance and generalizability.

**References**

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Adamo, F., Andria, G., Attivissimo, F., Lanzolla, A. M. L., & Spadavecchia, M. (2013). A comparative study on mother wavelet selection in ultrasound image denoising. *Measurement*, *46*(8), 2447-2456.

Biradar, S., Akkasaligar, P. T., & Biradar, S. (2022). Feature extraction and classification of digital kidney ultrasound images: a hybrid approach. *Pattern Recognition and Image Analysis*, *32*(2), 363-372.

Challenge, I. L. S. V. R. Olga russakovsky, jia deng, hao su, jonathan krause, sanjeev satheesh, sean ma, zhiheng huang, andrej karpathy, aditya khosla, michael bernstein, alexander c. berg, li fei-fei. 2014. *Computing Research Repository, Vol. abs/1409.0575*.

Chen, C. J., Pai, T. W., Hsu, H. H., Lee, C. H., Chen, K. S., & Chen, Y. C. (2020). Prediction of chronic kidney disease stages by renal ultrasound imaging. *Enterprise Information Systems*, *14*(2), 178-195.

Chollet, F., & others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.

Dozat, T. (2016). Incorporating nesterov momentum into adam.

Dyagilev, K., & Saria, S. (2016). Learning (predictive) risk scores in the presence of censoring due to interventions. *Machine Learning*, *102*(3), 323-348.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026-1034).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).

Kaur, G., & Singh, S. Image Quality Enhancement and Noise Reduction in Kidney Ultrasound Images.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kong, S., & Takatsuka, M. (2017, May). Hexpo: A vanishing-proof activation function. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 2562-2567). IEEE.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90.

Lee, L. H., Gao, Y., & Noble, J. A. (2021, June). Principled ultrasound data augmentation for classification of standard planes. In *International Conference on Information Processing in Medical Imaging* (pp. 729-741). Springer, Cham.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* (Vol. 30, No. 1, p. 3).

Meng, D., Zhang, L., Cao, G., Cao, W., Zhang, G., & Hu, B. (2017). Liver fibrosis classification based on transfer learning and FCNet for ultrasound images. *Ieee Access*, *5*, 5804-5810.

Misra, S., Jeon, S., Managuli, R., Lee, S., Kim, G., Lee, S., ... & Kim, C. (2021). Ensemble Transfer Learning of Elastography and B-mode Breast Ultrasound Images. *arXiv preprint arXiv:2102.08567*.

Peride, I., Rădulescu, D., Niculae, A., Ene, V., Bratu, O. G., & Checheriță, I. A. (2016). Value of ultrasound elastography in the diagnosis of native kidney fibrosis. *Medical ultrasonography*, *18*(3), 362-369.

Sharma, S., & Guleria, K. (2022, April). Deep learning models for image classification: comparison and applications. In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* (pp. 1733-1738). IEEE.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.

Tan, H. H., & Lim, K. H. (2019, June). Vanishing gradient mitigation with deep learning neural network optimization. In *2019 7th international conference on smart computing & communications (ICSCC)* (pp. 1-4). IEEE.

Torres, H. R., Queiros, S., Morais, P., Oliveira, B., Fonseca, J. C., & Vilaca, J. L. (2018). Kidney segmentation in ultrasound, magnetic resonance and computed tomography images: A systematic review. *Computer methods and programs in biomedicine*, *157*, 49-67.

Zhou, Z., Wang, Y., Guo, Y., Qi, Y., & Yu, J. (2019). Image quality improvement of hand-held ultrasound devices with a two-stage generative adversarial network. *IEEE Transactions on Biomedical Engineering*, *67*(1), 298-311.