Alexandria University

CSED21

Ahmad Abdallah Waheb (06)
Ahmed Mohamed Abdelhameed Elzeny (8)
Abdallah Mohamed Fawzy Elsaman (27)
Mohamed Salah Mahmoud Osman(42)
Mohamed Mohamed Abdel Hakim Mohamed(45)

MRNet: Deep-learning-assisted diagnosis for knee magnetic resonance imaging

Introduction

Magnetic resonance imaging (MRI) of the knee is the preferred method for diagnosing knee injuries. However, the interpretation of knee MRI is time-intensive and subject to diagnostic error and variability. An automated system for interpreting knee MRI could prioritize high-risk patients and assist clinicians in making diagnoses. Deep learning methods, in being able to automatically learn layers of features, are well suited for modeling the complex relationships between medical images and their interpretations. In this study, we developed a deep learning model for detecting general abnormalities and specific diagnoses (anterior cruciate ligament [ACL] tears and meniscal tears) on knee MRI exams. We then measured the effect of providing the model's predictions to clinical experts during interpretation.

Project Pipeline

Data loading: Due to the large data size we couldn't load the complete Dataset in the memory; we designed a Data Generator to load the data on the fly and perform random augmentation on the images on the exams.

Data pre-processing: Offline data augmentation could have a good impact on reproducing results and weights but it wasn't efficient because of the large data size. We performed our data augmentation and processing on the fly during the loading step which resulted in different augmentation each time an exam was loaded. Our augmentation consisted of random rotation between -25, 25 degrees, random shifting between -25, 25 pixels, and random horizontal flips with probability 50%. The augmentation routine was applied to each image in the exam independently from other images.

Data balancing: we tried different methods to balance the data. The first trial was to oversample the data with the minority class but its results were unsatisfactory because of redundant data, so we tried to augment them but no gain was reached. The best trial to balance data was to use class weights to apply it in the loss function when fitting the model.

Model Skeleton: We used a model skeleton with the feature extractor as a custom layer or a model and built a custom layer to call the feature extractor for each exam in the batch and apply the average pooling layer across the feature maps then max across the slices dimension. Then stack them in a NumPy array and apply them to a fully connected layer. Then, we generalize this model skeleton to all the CNNs used in training (Alexnet, VGG16, Resnet50, Inception V3).

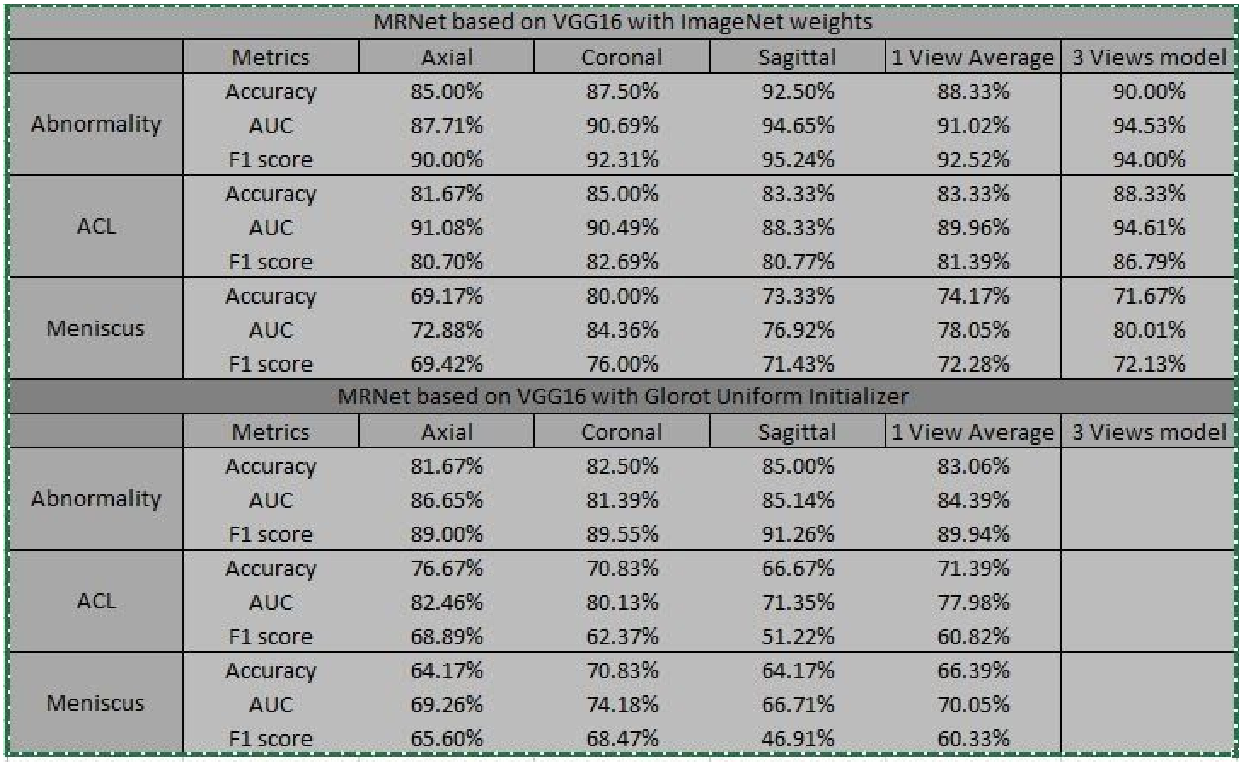
Model training: We have tried different optimizers like SGD, Adadelta, RMSprop and Adam. After All successful training trials were using Adam optimizer and a learning rate of (0.00001). All models of different angles and different diagnostics were trained each on a separate notebook on a different account to make use of time. The epochs count was different for each model according to depth and width of the model. For example, VGG is trained for nearly 50 epochs, inception 50-100 epochs, resnet for 10-20 epochs, Alexnet for 12-50 epochs.

Saving weights: We used a callback to save the weights after each epoch and used the saved weights to initialize the model and used cross-validation to choose the best set of weights.

Evaluation of 9 models and combining the predictions using Logistic regression (LR): we built an LR model to take the predictions of the 3 different angles and train the LR to weight the contribution of each angle to the final prediction and the model is generic and used for all CNNs , also we built LR data generator to predict the probabilities of 3 angles on the fly instead of saving and loading the predictions, we tried it in VGG16, but in the case of inception, the model stuck in GPU resources error. So we saved the predictions and used a modified version of the data generator to train the LR model.

Results and comparing the accuracy

VGG



MRNet based on VGG16 with ImageNet weights						
	Metrics	Axial	Coronal	Sagittal	1 View Average	3 Views model
Abnormality	Accuracy	85.00%	87.50%	92.50%	88.33%	90.00%
	AUC	87.71%	90.69%	94.65%	91.02%	94.53%
	F1 score	90.00%	92.31%	95.24%	92.52%	94.00%
ACL	Accuracy	81.67%	85.00%	83.33%	83.33%	88.33%
	AUC	91.08%	90.49%	88.33%	89.96%	94.61%
	F1 score	80.70%	82.69%	80.77%	81.39%	86.79%
Meniscus	Accuracy	69.17%	80.00%	73.33%	74.17%	71.67%
	AUC	72.88%	84.36%	76.92%	78.05%	80.01%
	F1 score	69.42%	76.00%	71.43%	72.28%	72.13%
MRNet based on VGG16 with Glorot Uniform Initializer						
	Metrics	Axial	Coronal	Sagittal	1 View Average	3 Views model
Abnormality	Accuracy	81.67%	82.50%	85.00%	83.06%	
	AUC	86.65%	81.39%	85.14%	84.39%	
	F1 score	89.00%	89.55%	91.26%	89.94%	
ACL	Accuracy	76.67%	70.83%	66.67%	71.39%	
	AUC	82.46%	80.13%	71.35%	77.98%	
	F1 score	68.89%	62.37%	51.22%	60.82%	
Meniscus	Accuracy	64.17%	70.83%	64.17%	66.39%	
	AUC	69.26%	74.18%	66.71%	70.05%	
	F1 score	65.60%	68.47%	46.91%	60.33%	

ResNet

		training		Validation	
		Accuracy %	AUC %	Accuracy %	AUC %
Abnormal	axial	93.9	98.6	78	48
	coronal	92.4	97.21	79.1	58
	sagittal	92.6	97.5	79.1	53
ACL	axial	90.1	97.37	53	48
	coronal	93.1	97.25	55.8	50.8
	sagittal	93.2	97.0	55	45
Meniscus	axial	90.82	96.72	52	54
	coronal	92.4	98.1	52.5	52.1
	sagittal	94.3	98.8	63	60

AlexNet

		Training		**Validation**		**Testing**	
		Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
Abnormal	Axial	95.81%	97.12%	83.34%	74.99%	77.29%	73.87%
	Coronal	96.31%	98.44%	85.00%	79.90%	85.70%	76.39%
	Sagittal	96.23%	98.28%	85.00%	86.42%	83.28%	76.42%
ACL	Axial	98.35%	98.82%	80.83%	88.52%	87.54%	80.32%
	Coronal	96.89%	97.80%	66.67%	67.85%	89.42%	77.50%
	Sagittal	97.53%	98.45%	63.34%	79.94%	92.22%	89.23%
Meniscus	Axial	94.51%	97.68%	61.67%	68.01%	78.28%	48.67%
	Coronal	93.28%	96.88%	65.35%	73.38%	55.66%	76.05%
	Sagittal	95.45%	97.60%	69.54%	75.06%	77.70%	73.66%

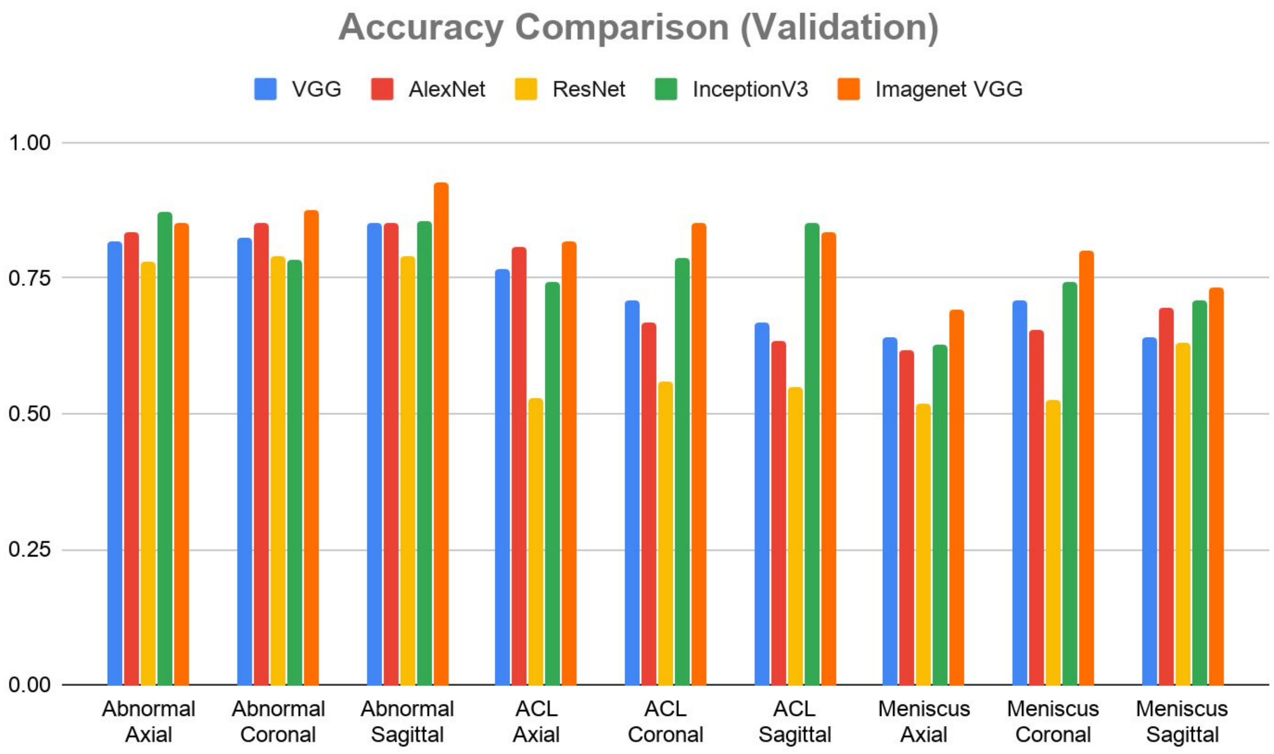
Inception

		Training		Validation	
		Accuracy	AUC	Accuracy	AUC
Abnormal	Axial	90.50%	94.73%	87.06%	87.75%
	Coronal	94.71%	97.81%	78.46%	69.09%
	Sagittal	92.68%	97.07%	85.48%	81.91%
ACL	Axial	97.96%	98.91%	74.41%	83.48%
	Coronal	88.38%	95.13%	78.81%	75.53%
	Sagittal	94.05%	97.64%	85.00%	74.01%
Meniscus	Axial	89.30%	94.26%	62.81%	61.43%
	Coronal	95.54%	97.71%	74.21%	73.69%
	Sagittal	71.04%	81.40%	70.83%	79.24%

LR model	Training accuracy	Training AUC	Validation Accuracy	Validation AUC
Abnormal	90.08%	95.29%	76.01%	86.75%
ACL	98.53%	99.55%	79.37%	61.77%
Meniscus	94.88%	99.03%	75.88%	61.84%

CONCLUSION

Comparing the results of the four models, we found out that the VGG model yields the best results


Accuracy Comparison (Validation)
VGG
AlexNet
ResNet
InceptionV3
Imagenet VGG
Abnormal Axial
Abnormal Coronal
Abnormal Sagittal
ACL Axial
ACL Coronal
ACL Sagittal
Meniscus Axial
Meniscus Coronal
Meniscus Sagittal