**Machine Learning for Engineering and Science Applications**
**Professor Dr. Ganapathy Krishnamurthi**
**Department of Engineering Design**
**Indian Institute of Technology Madras**
**Applications: Cardiac MRI Analysis - Tensorflow code walkthrough**

Hello, my name is Mahindra. I am a course TA and I am a Ph.D. student working with Dr Ganapathy Krishnamurthi. I will be continuing with the next session of where Ganapathy sir left. So I will be describing about the dataset.

(Refer Slide Time 0:30)

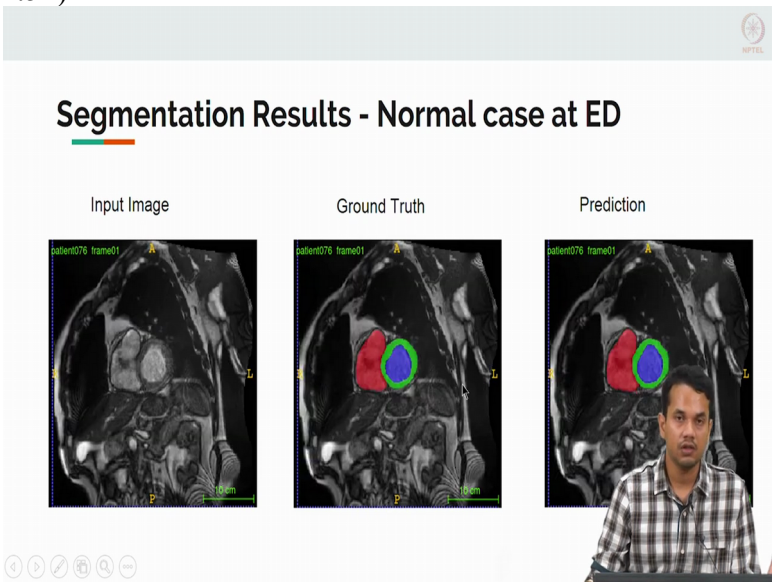

For our challenge, we used 3 publicly available datasets. They are the ACDC challenge dataset, the STACOM and the Kaggle Datascience challenge dataset. In the ACDC dataset, we were provided with 100 cine MRI cases and for the task of segmentation, we were given the ground truth annotations for the left ventricle, right ventricle and the myocardium. And these annotations were provided End systole and End diastole and also the MR case has involved 5 groups of patients, namely the normal, the dilated cardiomyopathy and hypertrophic cardiomyopathy and myocardial infarction and abnormal right ventricle.

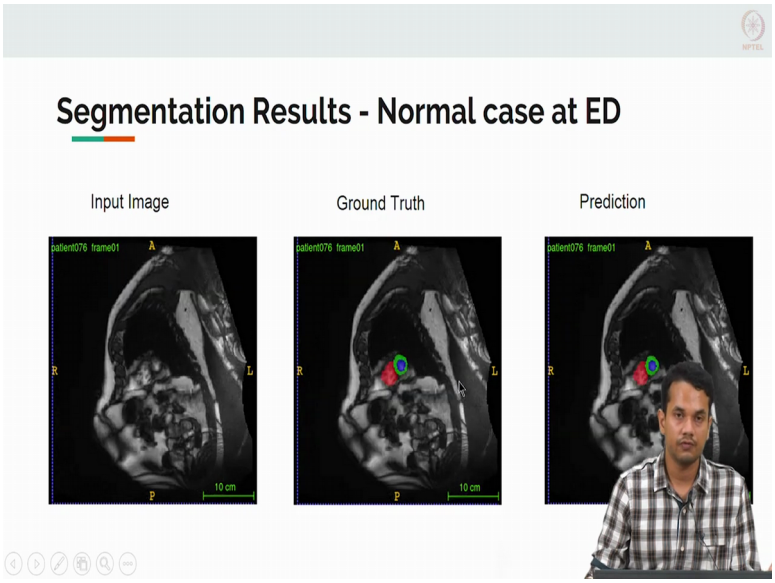
For validating our algorithm, they provided us with 50 cine MRI cases. Similarly in the STACOM data, they also provide us with 100 cine MRI cases and 2 groups of patient categories were involved and the testing set was even 100 here. This Kaggle Datascience Bowl is an annual

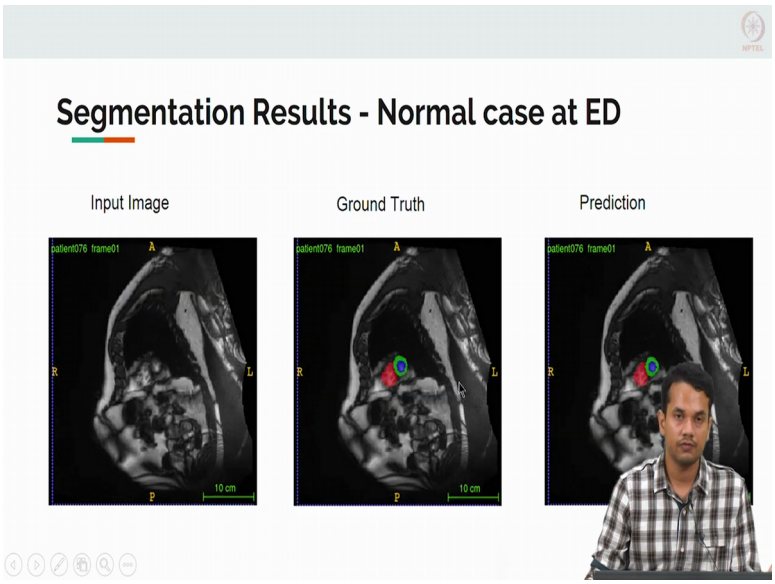
challenge hosted on various medical problems. So in 2016, they hosted a challenge for predicting the volumes of the ventricles in systole and diastole.

Their training set involved 700 cine MRI cases and here no annotations were provided but instead only the reference volumes that systole and diastole provided.

(Refer Slide Time 1:54)

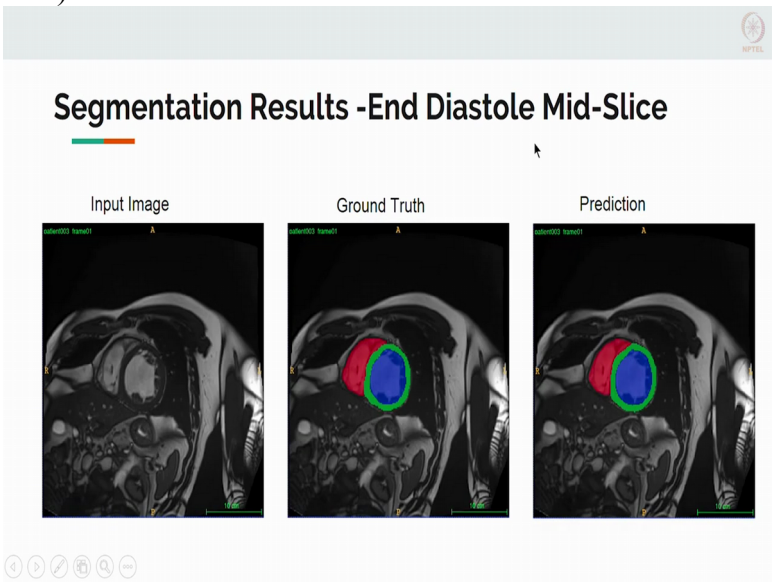




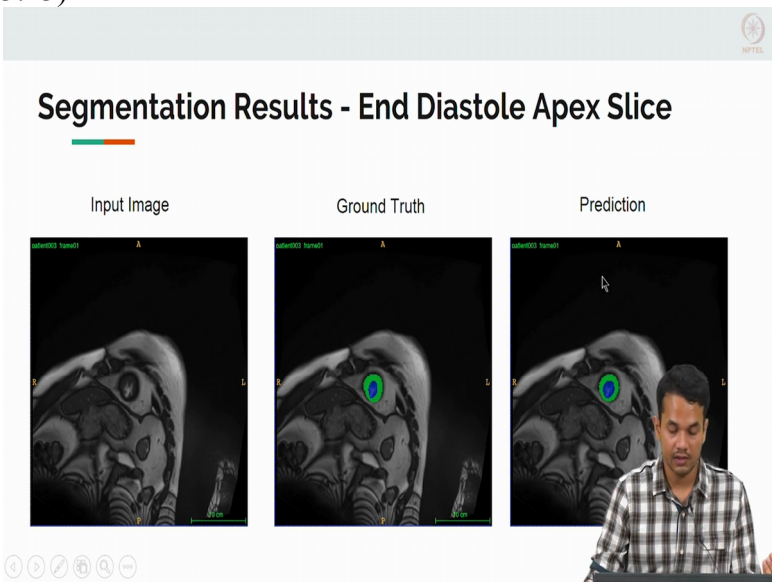

So I hope the visual visual description of the results I will be presenting to you of our model. So you could see here, this is our input image and this is the ground truth. Ground truth means the segmentation contours drawn by the radiologist. And this is the prediction from a model. You can see that at a particular case of normal and at End diastole phase, the slices are taken from the apex till the base. You can see the our prediction model has very close similarity with the the ground truth.

(Refer Slide Time 2:37)



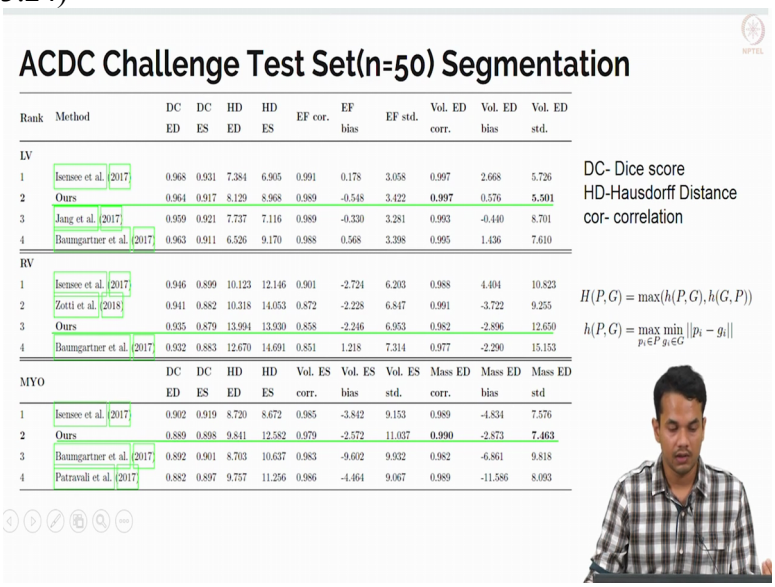
And even in the systole case, I am showing the results. The One of the challenges in cardiac segmentation is the segmentation at the topmost and the bottommost slices of the heart, mainly the basal slice which are close to the arteries and the epical slice which are close to the end of the heart. So these are very difficult regions to segment. We can see here. These are mispredictions seen here. The, our model does very well in predicting the mid-slice region of the heart.

(Refer Slide Time 3:13)




We could see here in the prediction and the ground truth. And when it comes to apex slices, because of the small region, it is quite ambiguous sometimes to clearly demarcate the epical assertions.

(Refer Slide Time 3:24)

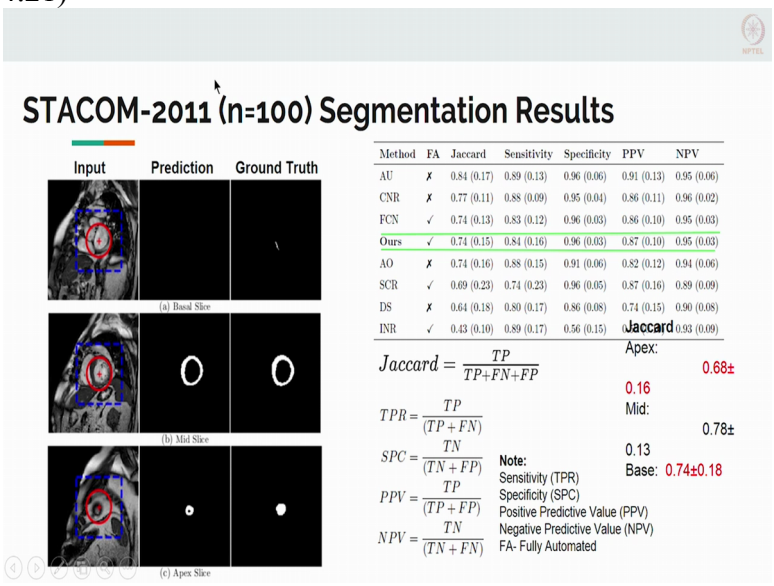



| Rank | Method | DC ED | DC ES | HD ED | HD ES | EF cor. | EF bias | EF std. | Vol. ED corr. | Vol. ED bias | Vol. ED std. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LV | | | | | | | | | | | |
| 1 | Isensee et al. (2017) | 0.968 | 0.931 | 7.384 | 6.905 | 0.991 | 0.178 | 3.058 | 0.997 | 2.668 | 5.726 |
| 2 | Ours | 0.964 | 0.917 | 8.129 | 8.968 | 0.989 | -0.548 | 3.422 | **0.997** | 0.576 | **5.501** |
| 3 | Jang et al. (2017) | 0.959 | 0.921 | 7.737 | 7.116 | 0.989 | -0.330 | 3.281 | 0.993 | -0.440 | 8.701 |
| 4 | Baumgartner et al. (2017) | 0.963 | 0.911 | 6.526 | 9.170 | 0.988 | 0.568 | 3.398 | 0.995 | 1.436 | 7.610 |
| RV | | | | | | | | | | | |
| 1 | Isensee et al. (2017) | 0.946 | 0.899 | 10.123 | 12.146 | 0.901 | -2.724 | 6.203 | 0.988 | 4.404 | 10.823 |
| 2 | Zotti et al. (2018) | 0.941 | 0.882 | 10.318 | 14.053 | 0.872 | -2.228 | 6.847 | 0.991 | -3.722 | 9.255 |
| 3 | Ours | 0.935 | 0.879 | 13.994 | 13.930 | 0.858 | -2.246 | 6.953 | 0.982 | -2.896 | 12.650 |
| 4 | Baumgartner et al. (2017) | 0.932 | 0.883 | 12.670 | 14.691 | 0.851 | 1.218 | 7.314 | 0.977 | -2.290 | 15.153 |

| MYO | | DC ED | DC ES | HD ED | HD ES | Vol. ES corr. | Vol. ES bias | Vol. ES std. | Mass ED corr. | Mass ED bias | Mass ED std |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Isensee et al. (2017) | 0.902 | 0.919 | 8.720 | 8.672 | 0.985 | -3.842 | 9.153 | 0.989 | -4.834 | 7.576 |
| 2 | Ours | 0.889 | 0.898 | 9.841 | 12.582 | 0.979 | -2.572 | 11.037 | **0.990** | -2.873 | **7.463** |
| 3 | Baumgartner et al. (2017) | 0.892 | 0.901 | 8.703 | 10.637 | 0.983 | -9.602 | 9.932 | 0.982 | -6.861 | 9.818 |
| 4 | Patravali et al. (2017) | 0.882 | 0.897 | 9.757 | 11.256 | 0.986 | -4.464 | 9.067 | 0.989 | -11.586 | 8.093 |

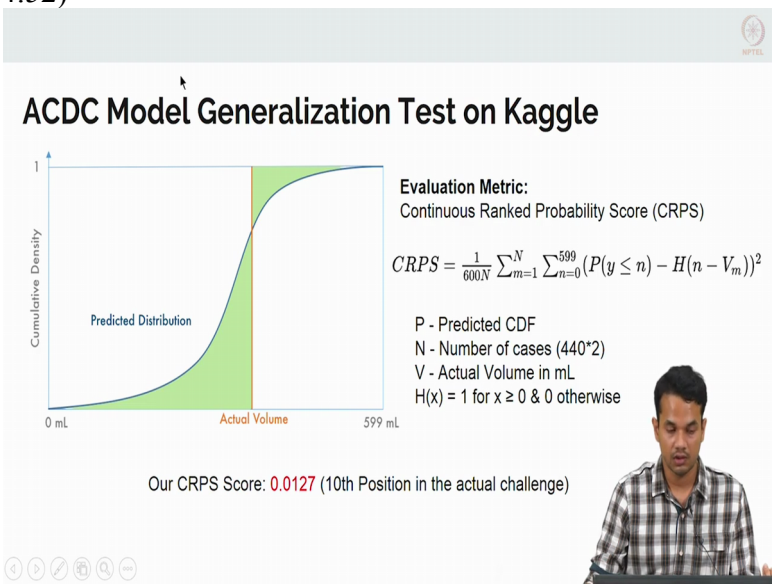
In the ACDC challenge test set, our model gave a comparative competitive result and we stood 2nd in the challenge and the matrix used were both clinical as well as clean geometric matrix. The clinical matrix included the ejection fraction, the volume set, ED and ES. And the geometrical matrix were dice score and Hausdorff distance. So using this matrix, they evaluated our segmentation performance. We could see that Jack in the left ventricle we stood a rank of 2 and in the right ventricle, we had a rank of 3 and in the myocardial segmentation, we had a rank of 2. Our overall ranking was 2nd in this challenge and the results are compared with the other participants of the challenge.

(Refer Slide Time 4:21)




Even in the STACOM challenge, we achieved a fairly competitive result. Our method was fully automatic and we were on state of Jack with the other techniques. The matrix used was for the jaccard index. We could clearly see here that even our model had slightly lower predictions for the apex and the base slices. Whereas in the mid-slice, the predictions were really good.

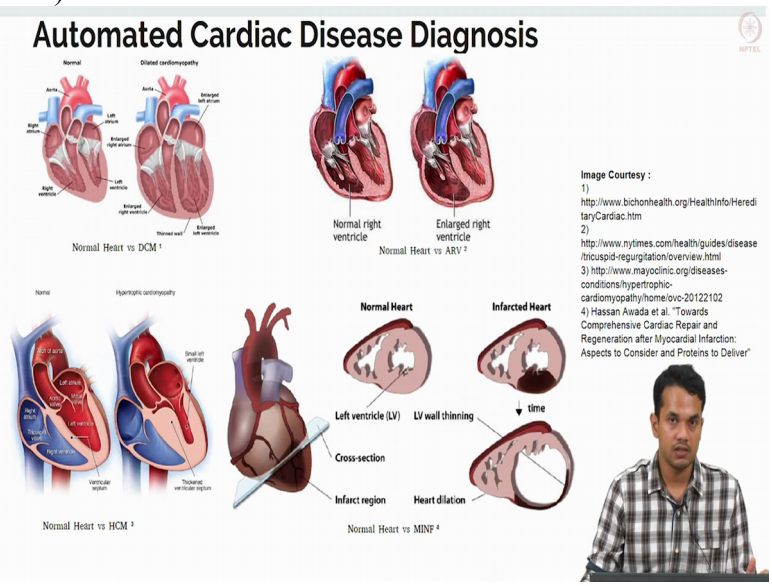
(Refer Slide Time 4:52)




In case of Kaggle dataset, we did not use any evaluation dataset from the Kaggle. Instead, we used our ACDC model, trained on the ACDC dataset for directly testing on the Kaggle test set. They used a continuous ranked probability score which is defined in the formula below. This

gives us a score based on the predicted distribution of the actual volume. So our score was around 0.0127 which gave us a position of 10th in the actual challenge.

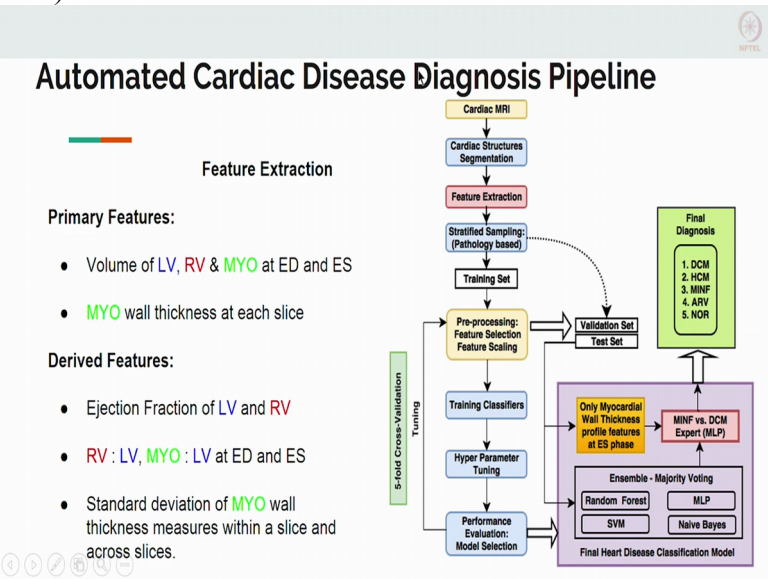
(Refer Slide Time 5:29)




So I will be discussing the next part of our work that is the automated diagnosis. In the ACDC challenge the task involved 2 steps, one is the segmentation as well as the automated cardiac diagnosis. Once the segmentations were got from the cardiac MRI, we used them to predict the disease of the heart. As I previously mentioned in the dataset, there were 5 patient categories. They were described as shown in the figure now. So in the normal heart, is given here and it is compared with the dilated cardiomyopathy.

We can see in the DCM, the walls are thin and the ventricles are dilated. In the abnormal right ventricle, the ventricle regions are enlarged here. In the hypertrophic cardiomyopathy, the walls of the myocardium are thick and in case of infarction, certain regions of the myocardial segments get thin compared to the rest of the myocardium region. So these were the 5 patient categories given in the ACDC dataset. The objective was to develop automated algorithm which does the classification.

(Refer Slide Time 6:49)




So this is our diagnosis pipeline. So any machine learning algorithm has certain steps. In the first very step here was extracting the features. So from the segmentation, we extracted the features. We basically categorised the features into 2 groups, primary and the derived. The primary features are like the volume of the ventricles and the mass of the myocardium and also was thickness of the myocardium at each slice.

And the derived features are something you derive from the primary features like the ejection fraction of the left ventricle and the right ventricle, the ratios of the volumes of the ventricles and estimating the standard deviation of the myocardial wall thickness measures within a slice and across the slice. So the pipeline is explained here as follows. The first step involves from the cardiac MRI in the segmentation, we used our segmentation model, to get the segmentations from the cardiac MRI, we used feature extractions module to extract features.

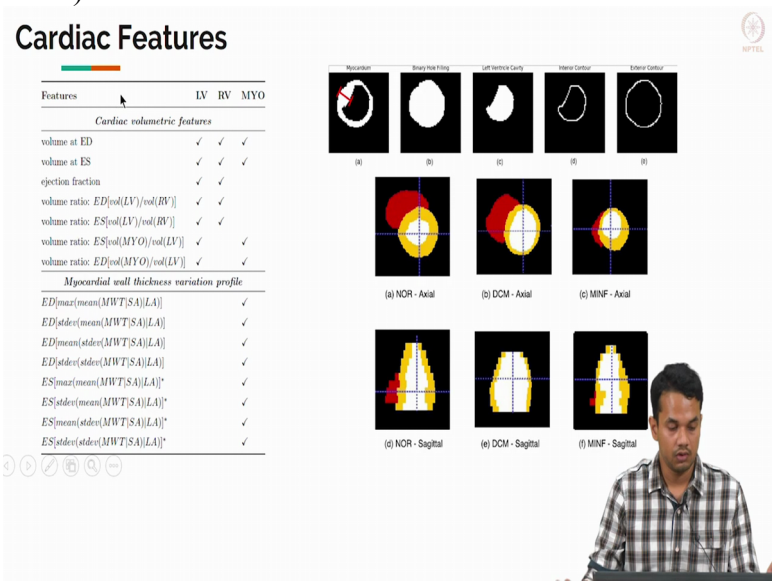
The features extracted are described in the next slide and we classify we group the training set into training set and validation set. This pit was done for our model selection. Mainly we had to pick the best classifier for training our algorithm to classified to 5 classes as well as for running the hyper parameters. So the preprocessing involved feature selection as well as features scaling and training the classifiers and tuning the hyper parameters and dipping the model selection. So we use fivefold cross validation for model selection as well as feature selection and fee and hyper parameter tuning.

Based on the fivefold cross validation, we found that Jack ensemble system desk very good in predicting the disease group, so we developed a two-stage model. In the first stage, it was a aggregation of 4 classifiers. It is classifier was trained independently on all the 5 patient categories using all the features and we found that these 4 classifiers gave the very best results individually. So we grouped them and each classifier output was taken and a majority vote was done to get the final prediction from the first stage of the diagnosis.

After the first stage of diagnosis, we passed to the 2nd stage. We observed that in certain patient categories, like in the infarction and the dilated cardiomyopathy, it is a closely related groups. It was quite difficult to discern the disease category. So we had to develop a expert classifier. We found out that MLP does the best job for this task and it was trained on a subset of features used for training the these 4 classifiers. We found that the myocardial wall thickness profile features at end systole were sufficient to properly segregate between these 2 classes.

So the 2 stage implementation is something like this. We have the first stage to get 5 class prediction and in the 2nd stage, we have a refinement to correct some of the misclassifications. So the final classification was one of the 5 categories here.

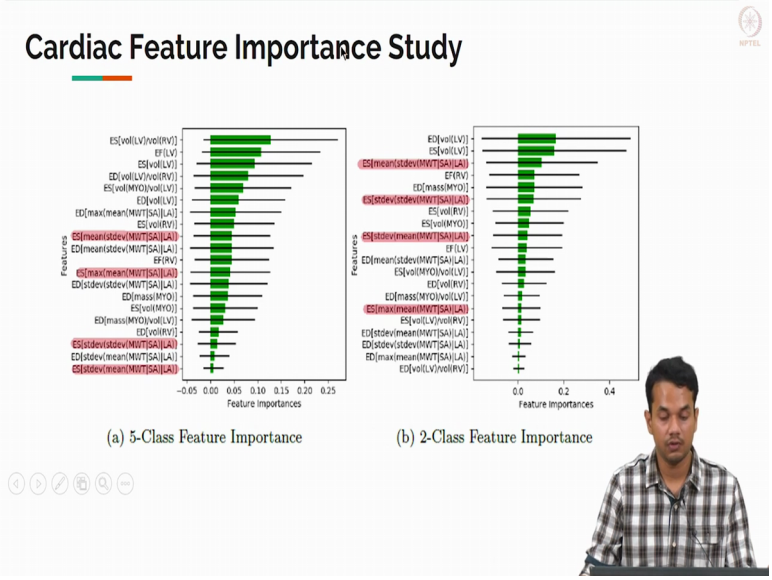
(Refer Slide Time 10:19)




So this slide lists some of the features used for our training. So as I told, the volume features, the ejection fraction, the volume ratios and the myocardial wall thickness variation profile was

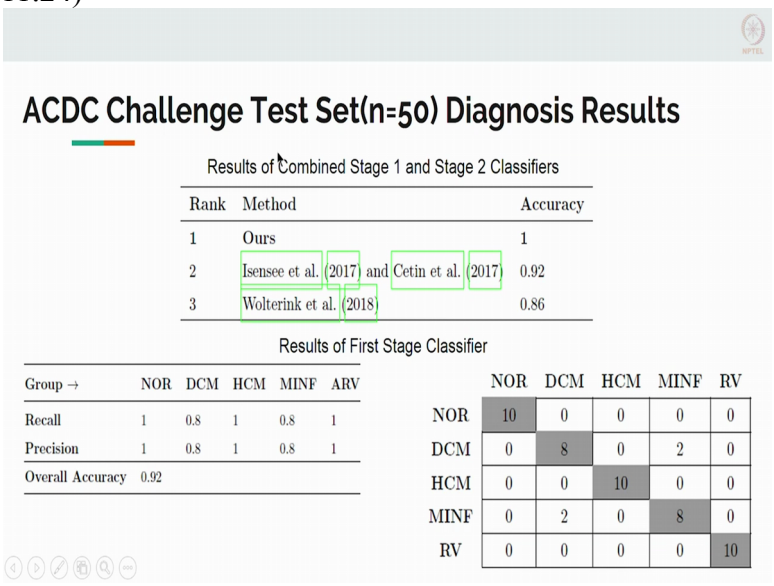
captured using various pa methods like finding the Max of the mean of the standard deviation and the variation across the short axis, long axis.

(Refer Slide Time 10:45)



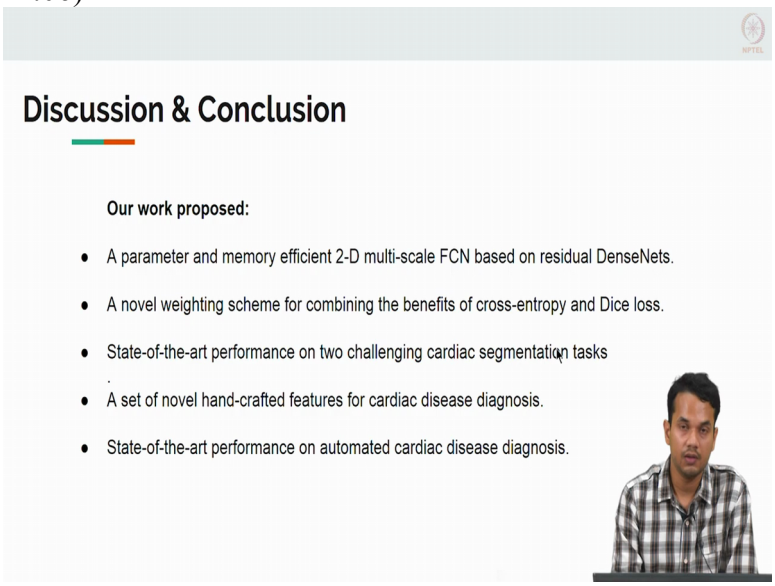

We also use random forest to select features. We can use random forest classifier for doing feature selection. So this is our one use case of using random forest classifier. So as I told, we had a two-stage diagnosis. In the 2nd stage, we had to figure out what are the relevant features for grouping the patient cases within infarction and the DCM. So we observed that random forest showed us, these are the important features at the End systole phase which are sufficient to segregate into 2 classes.

(Refer Slide Time 11:24)



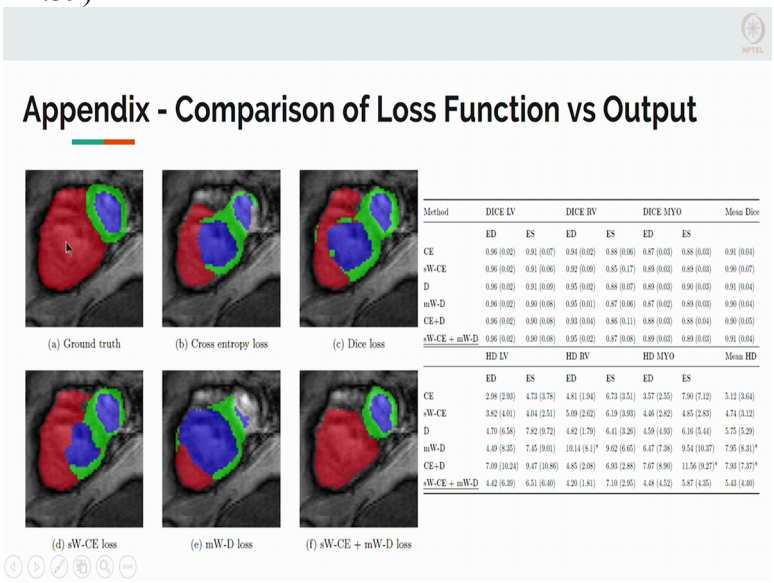

The challenge, our method achieved accuracy of 100% and this is attributed to our two-stage classification. I can clearly show you how this method was effective. So in the first stage, our classifier gave an overall accuracy of 92%, we could see there were misclassification in the DCM and MINF. Even in the confusion matrix, we could see some of the groups were most classified among DCM and MINF only. So up on the 2$^{nd}$ stage, all this misclassifications were gutted and we are able to achieve a 100% accuracy.

(Refer Slide Time 12:06)



So the paper concludes as follows. We actually developed a very parameters and memory efficient 2-D multi-scale FCN based on residual DenseNets. And we showed our novelness in weighting scheme used for combining the cross entropy and dice loss. We achieved state-of-the-art performance onto challenging cardiac segmentation tasks. We also handcrafted a novel features for cardio disease diagnosis. And we achieved a state-of-the-art performance on automated cardiac disease diagnosis.

(Refer Slide Time: 12:39)

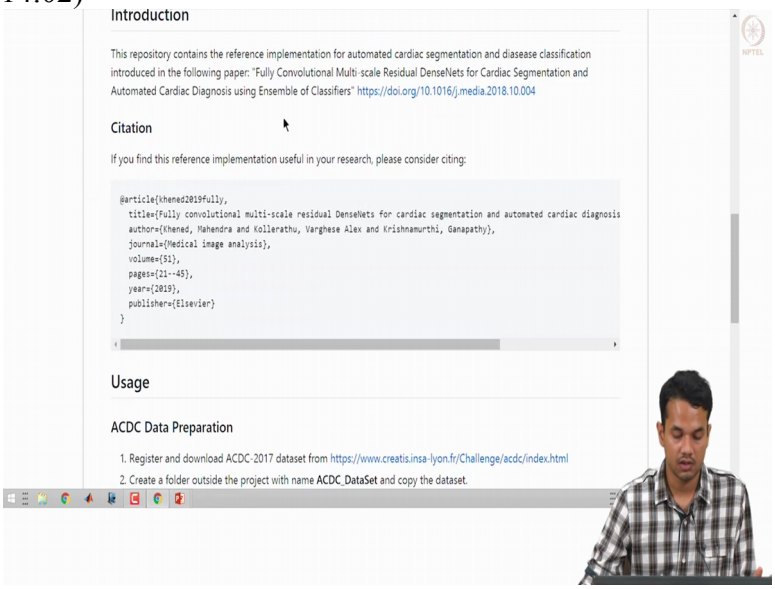



I would like to add one more point here. Our loss function was come compared to other standard loss, gave better visual appearance. We could clearly see here. These are the standard loss like cross entropy, weighted cross entropy, dice loss and weighting scheme used by our method. So we could see that when compared to ground truth, our model had a better visual appearance compared to other loss functions. So hence gave a better performance compared to other losses.

(Refer Slide Time 13:15)

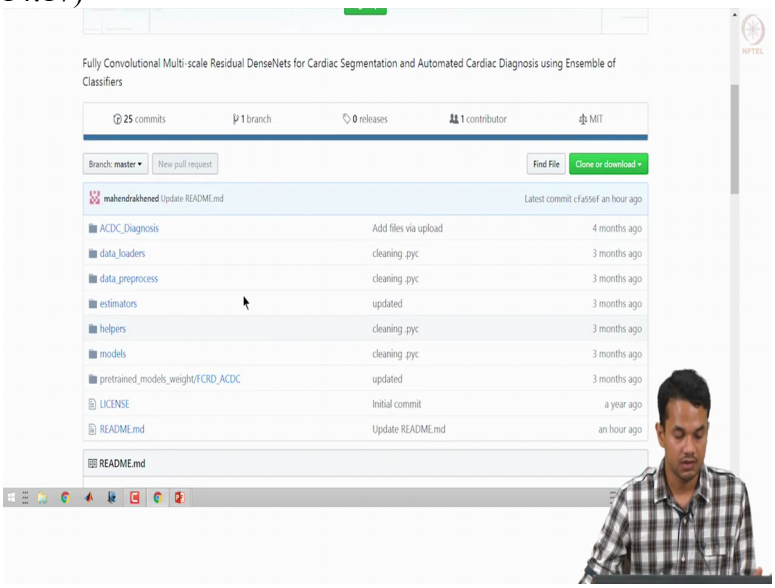

We have uploaded our source code online so that people can take benefit of it and I urge you to look into this code because many examples in tensor flow has been used a scale on has been used. This link would be posted in the announcement section. So please look into it and briefly gave a walk-through of the code so that you can find it easier to run in your system. So most of the codes these days are up with the research papers are coming up with the codes and they are uploaded in github. Github is a common platform where many projects can be developed in collaboration in it can be shared to the world.

(Refer Slide Time 14:02)



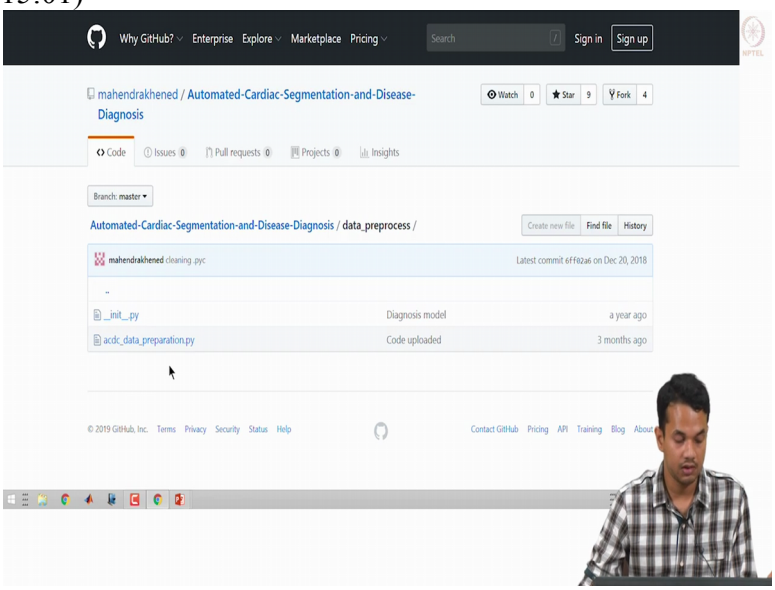
So our paper's code is put up in the github profile. The instructions for using the code has also been put up here and the method to train and test and validate is also given here.

(Refer Slide Time 14:17)




I will just give you a overview of how this code is being organised. So basically in any deep CNN work involves preparing the data, loading the data, preparing the model, the neural network architecture and developing some estimators for analysing while training like the loss function, the matrix for finding out the accuracy of the model. These are the groups mod these are the models in our code. To start off, we have to preprocess the data.

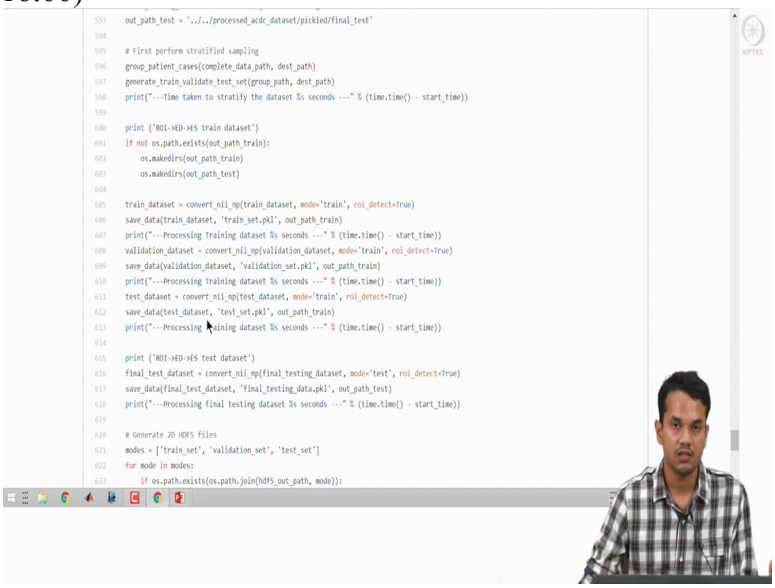
(Refer Slide Time 15:01)



This model, data preprocess has all the codes for preprocessing the raw data. When I say this cardiac data, it comes in a nifty format. These are very specific medical image formats which might not be efficient to directly process for neural network. So we need to actually extract the data jack extract the data, extract some of the information, preprocess them, all this have been given in this source file. All these are self-explanatory, I urge you to walk through it.

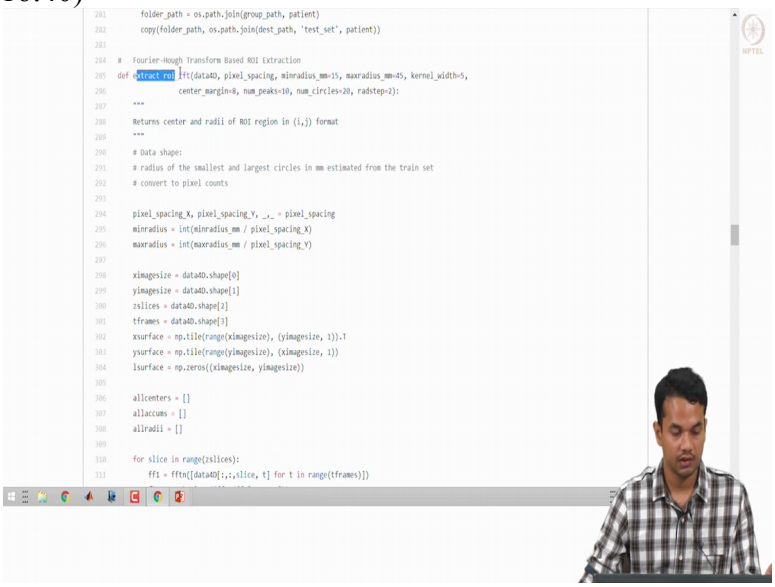
I will just give you an intro onto what this module does. Basically, the medical images are numpy formats, no nifty format and this need to be converted into numerical Python format, numpy. So in numpy we can do various mathematical operations for easy and faster processing.

(Refer Slide Time 16:00)



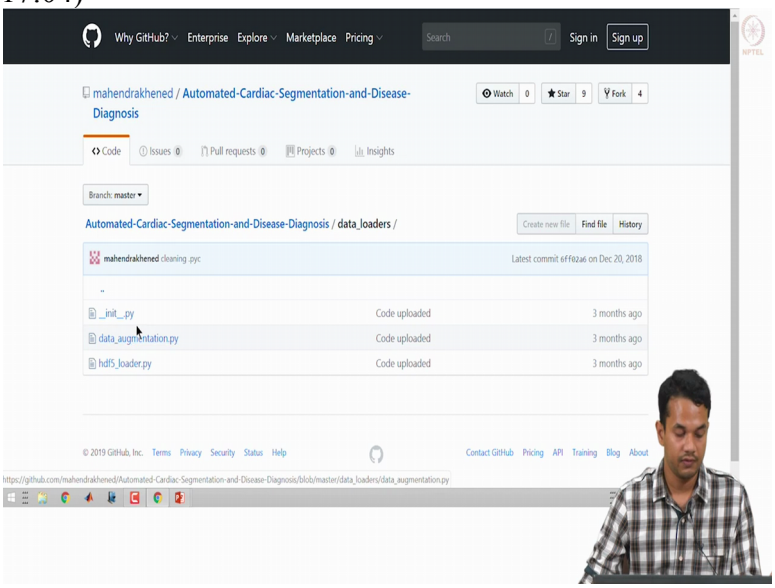

So basically the cardiac dataset as I explained, it is 4D volume or 4D stack. So at each phase of the cardiac, is a 3-D volume and each phase is a time point. So it is aggregation of time series of volume data. So we need to extract relevant slices with annotations are there and these slices need to be converted into suitable format. And this is, this code does that. Basically, we also need to preprocess to extract the ROA Centre.

(Refer Slide Time 16:40)



The code for ROA Centre prediction is in this file, the extract roi fft based methods. The codes are self-explanatory, you can look into it and reading and writing. And also one more thing is that these files have to be converted into proper formats like hp file format or hdfi file format for faster processing.

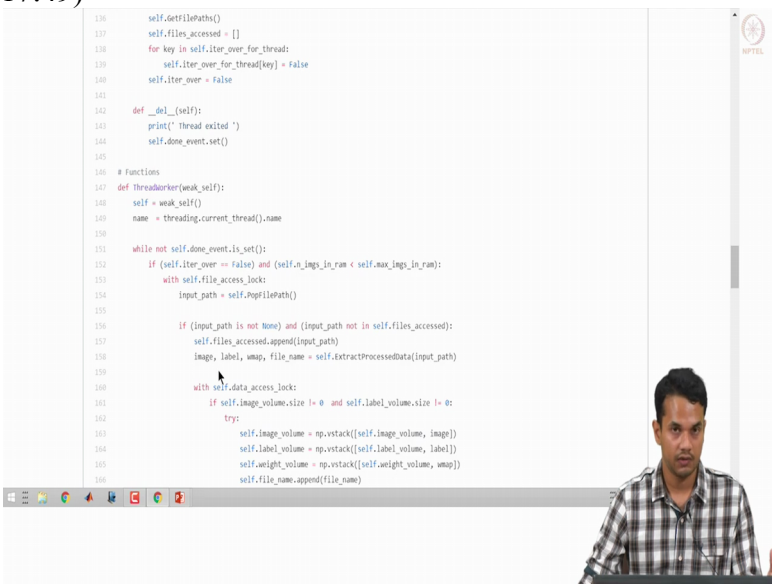
(Refer Slide Time 17:04)




The 2nd step here is preparation of the data loading. As I previously discussed, most of the neural networks are trained on a batches of the data and to enhance the dataset, even the augmentations are performed. This module does two task. When you need to read the network with data, it takes

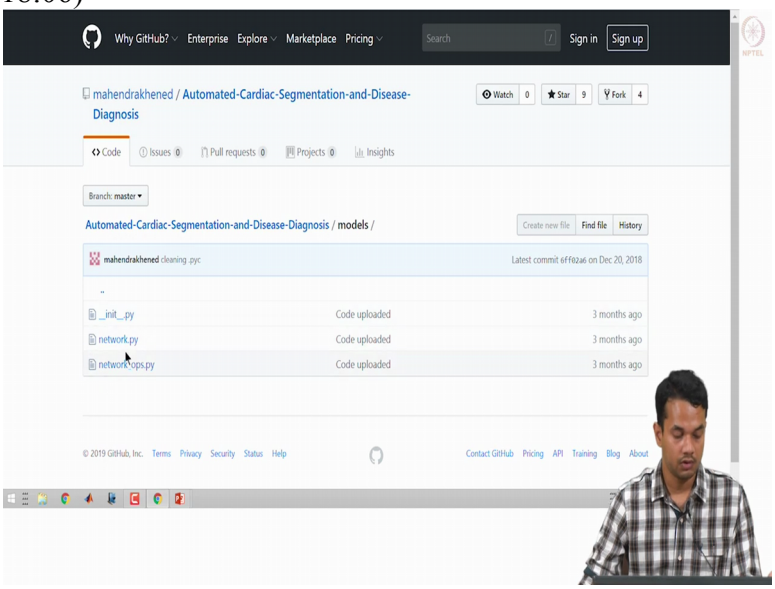
relevant files from the pre-processing and the preprocessing like the augmentation and the permissions are done in this file and also the normalisation.

(Refer Slide Time 17:49)




The loader basically from the preprocessed files, it extract the relevant images, prepares batches and these batches are passed through preprocessing module where the augmentation such as defamation, rotations are done, also the normalisation is done when these are fed to the neutral network.

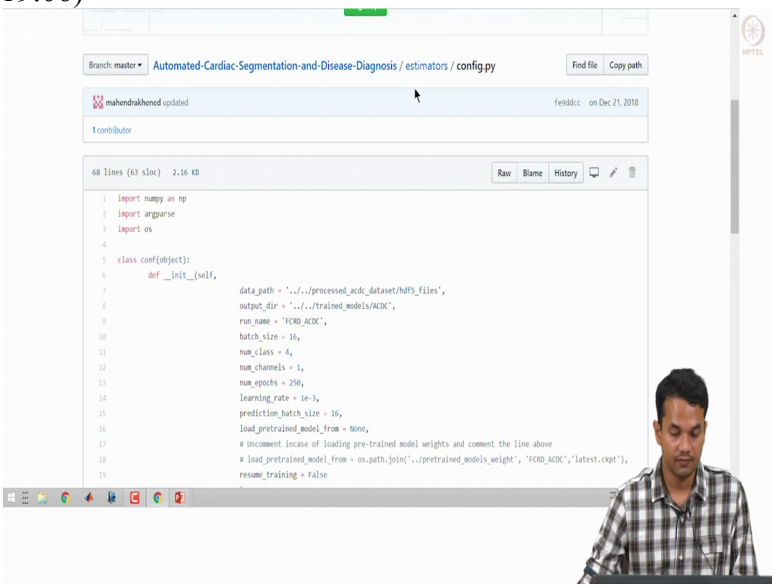
(Refer Slide Time 18:06)

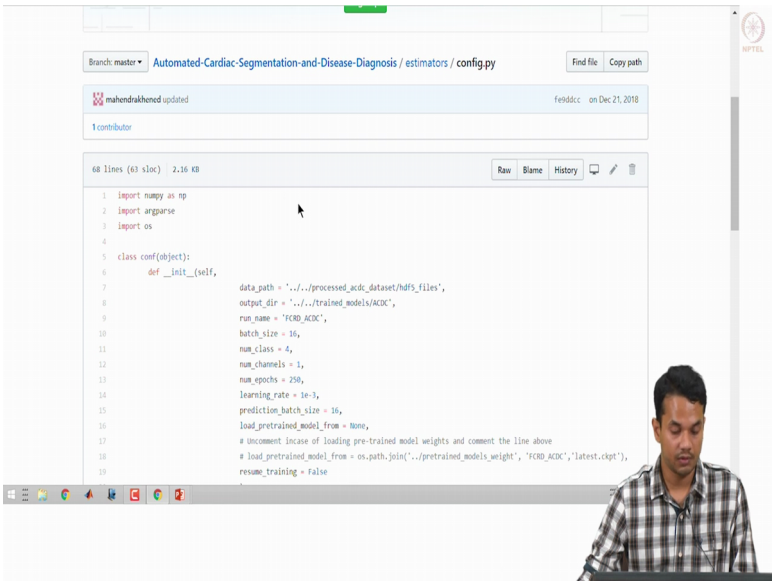


And in this model module, we define most of the tensor flows architecture. We have used tensor flow slope to build our network. So this is upper object where the where we define our neural network architecture. So we develop everything using tensor flow layers module and the lower primitive functions are developed in this file called network.ops. So where we develop basic building blocks for building our neural network and all these things, you can look into it for further understanding.

So how the dense blocks are developed, how the layers are defined, all these things you can look into these 2 files. Once the models are developed, we need to have a proper estimator.

(Refer Slide Time 19:06)

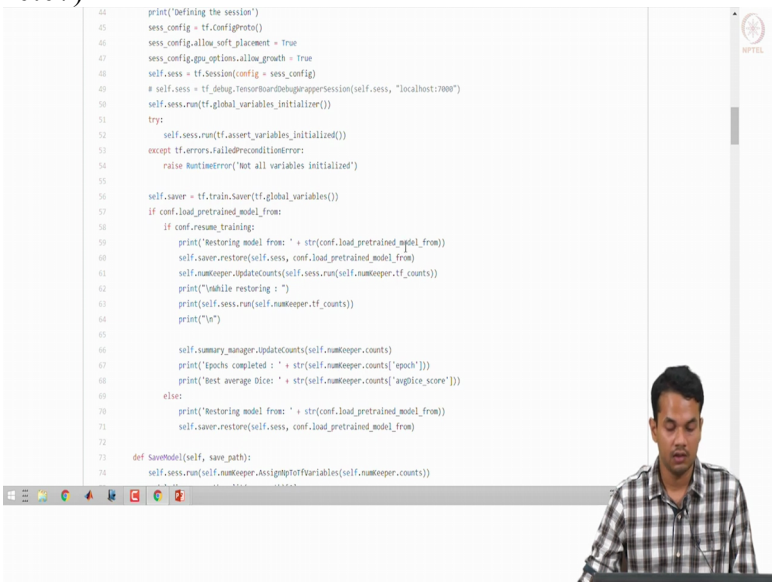


The estimator is something like which which is a bridge between the data loading as well as the model as well as estimating the statistics while training is happening. So here in the estimator, there is a config file and the train.pui file and the estimator file. The config file is file where we consider some of the hyper parameters like what is the batch size, what are the number of classes, number of epochs to be trained, what is the learning rate, where is the data located, all these things are configured in the config file.

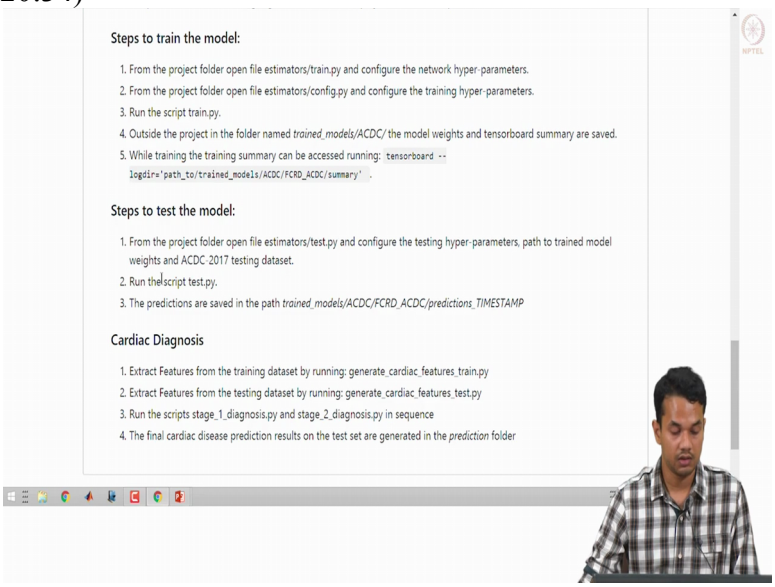
After the configuration is done, we have the train.pui file where we configure the model architecture, model hyper parameters like the growth rate, number of layers and the rate decay, dropout, all these things are configured in the train.pui. And this is the main file which we need to run for, to initiate the training procedure.

(Refer Slide Time 20:07)



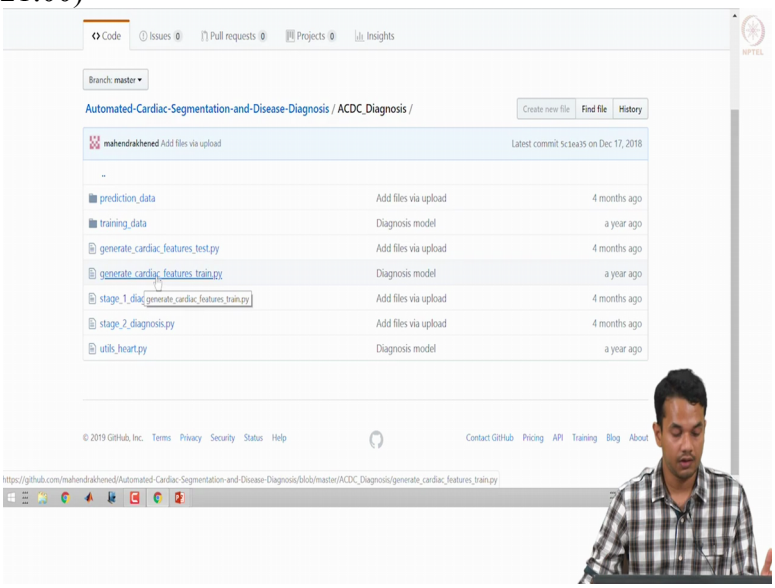

This is the estimator file where we estimate while training, what is the loss, how was the training progressing and also tensor board functions are implemented in this file. Tensor board is another tool which you can use it to visualise, to see how the training is progressing. You can see live updates of the loss as well as how the live predictions are happening on the training and validation set.

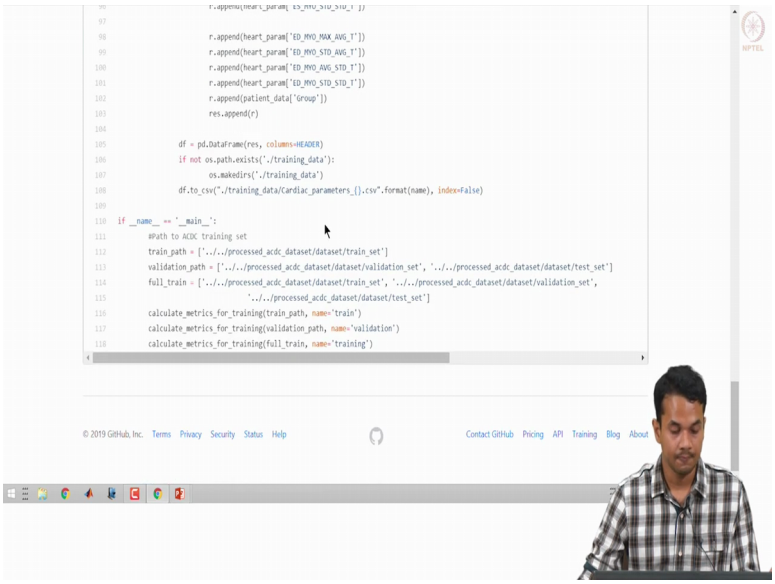
(Refer Slide Time 20:34)




Once the segmentation, we can use our testing module to test our segmentation results. We can also upload our test results in the website of the challenge, where they are evaluating our model. The 2nd part is the cardiac diagnosis and walking through this code.

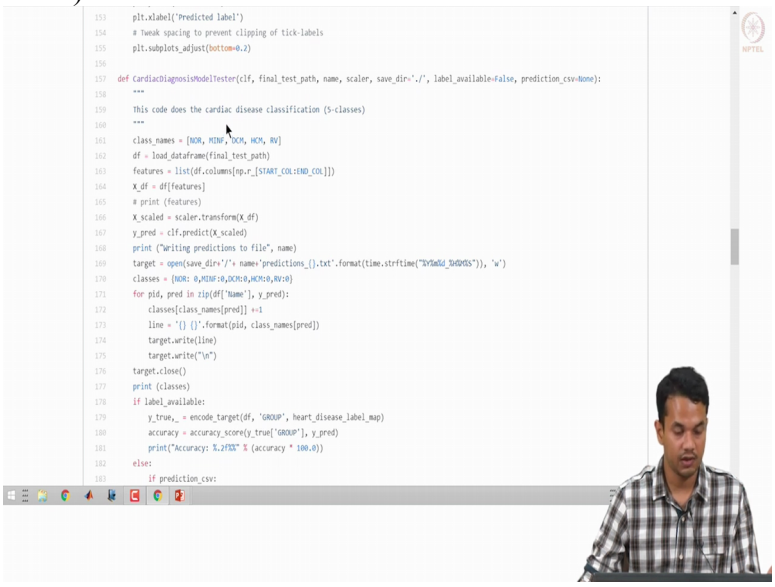
(Refer Slide Time 21:00)






Once the segmentations are done, we can run this generate.cardiac features file, this file has functions to read the segmentation masks. From the segmentation mask various features as described in the previous slides have are extracted and these are saved in a csv file. This csv file is used for training our classifier. In the stage 1 and the stage 2 diagnosis, we have classifier selection using cross validation studies. So basically we have used all scalan functions to implement our classifiers here.

(Refer Slide Time 21:46)




So what we do is we read the CSV file and we do fivefold cross validation study and we do the classifier selection and training of the model on the training set. Once the training is done, we

can use the trained model on the testing set. The stage 2 diagnosis does the similar operation but here the problem is limited to diagnosis or the correction between the DC (())(22:10) MINF cases. Once this file is run, we get the final prediction results in the prediction folder where we get the prediction of each cine MRI volume as either normal or anyone of the pathologies of the heart. You can check this model by running in your system and you can also get a feel of by tuning these parameters or playing around with the classifiers. You can upload in the challenge website and get a feedback of whatever results you have got. Thank you.