Group 5: Samuel 1005142 , Matthew 1004263

Topic: Triage Classification

Problem and Objective

In emergency departments (ED) at hospitals, triage nurses facilitate patient flow safely, ensuring timely diagnosis and management. However, lack of triage nurses could cause patient flow to bottleneck. Automating triage classification on incoming patients would help elevate workload on nurses.

Datasets:

For this project, we used 2 datasets that are related to triage.

The first is : (Hospital Triage and Patient History from Kaggle)
https://www.kaggle.com/datasets/maalona/hospital-triage-and-patient-history-data
, which includes patients symptoms and conditions during triage as well as their triage class (esi score, 5 classes).

The second is : (ecg image data)
https://www.kaggle.com/datasets/erhmrai/ecg-image-data/code
, which contains electrocardiogram (ECG) images of patients and their ecg type (6 classes).

Both are relevant to triage as nurses decide what priority to give patients based on their symptoms/conditions or ECG readings.

Model for first dataset:

For the first dataset (Hospital Triage and Patient History), we first approached by creating 2 models in the Triage folder in Triage.ipynb. The models are SmallModel and PaperModel.

The dataset consists of 972 features and 560,485 samples.
For both models, we trained taking in 970 features for the initial layer. These features include patients' details such as age, gender, ethnicity, race, religion, medical conditions, symptoms, and vital readings. Covering all binary, numeric and categorical features.

For SmallModel, 5 linear layers with ReLU activations are used together with Adam optimizer and cross entropy loss function. With a learning rate of 0.001 and no weight decay, results of Train Accuracy: 0.8512 and Test Accuracy: 0.8518 were obtained. (Other noticeable hyperparameters were [lr= 0.005, weight_decay= 0.001], and [lr= 0.01, weight_decay= 0.005].

For PaperModel, we used only 4 linear layers instead with ReLU activations which performed worse. Using the RMSprop optimizer with learning rate of 1e-3 with cross entropy loss instead improved slightly, but still performed badly with results of Train Accuracy: 0.2979 and Test Accuracy: 0.2950.

Even with SmallModel performing its best with test accuracy 0.8518, the more complex and pretrained models in pytorch such as resnet50 still outperforms our model with accuracy > 90%.

Next we filtered the dataset to try and train a model that is able specifically triage for chest pain patients, this is in the pre-processed.csv excel. Removing a lot of the 970 features, until only 51 were left, and trained the same models in Triage_proccessed.ipynb.

What we noticed was that the train and test accuracies were slightly lower instead with Train Accuracy: 0.8108 and Test Accuracy: 0.8104 for SmallModel. We believe that the decrease in accuracy was due to the removal of patient particulars kind of details in the features such as ethnicity, gender, race etc. And these features were actually key with an amount of weight to them that have effect over a patient's triage classification. Thus these features should be included for future models.

Following the lines of narrowing down on triaging chest pain type patients, we move on to the second dataset of ECG images which chest pain conditions normally require an ECG scan for more accurate triage.

Model for second dataset:

For the ECG images, the color of the different categories of ECG were different which resulted in the model being 100% accurate, as such we grayscale the images before turning into black and white. Example of image type (figure 1).

Figure 1

For the model, we used yolov7 from https://github.com/WongKinYiu/yolov7 to experiment with the dataset. The model can use transfer learning with the data and weights from the .pt files in /weights/. With optimizers type as Adam or SGD. Loss function of BCEWithLogitsLoss is used. Running the files train.py and test.py on the image data, our results can be found in /runs/train/exp. With the model and the best epoch and state recorded as .pt files. Running for 20 epochs, the model performs with mean Average Precision of 0.88.

Below are figures of the training results (figure 2):

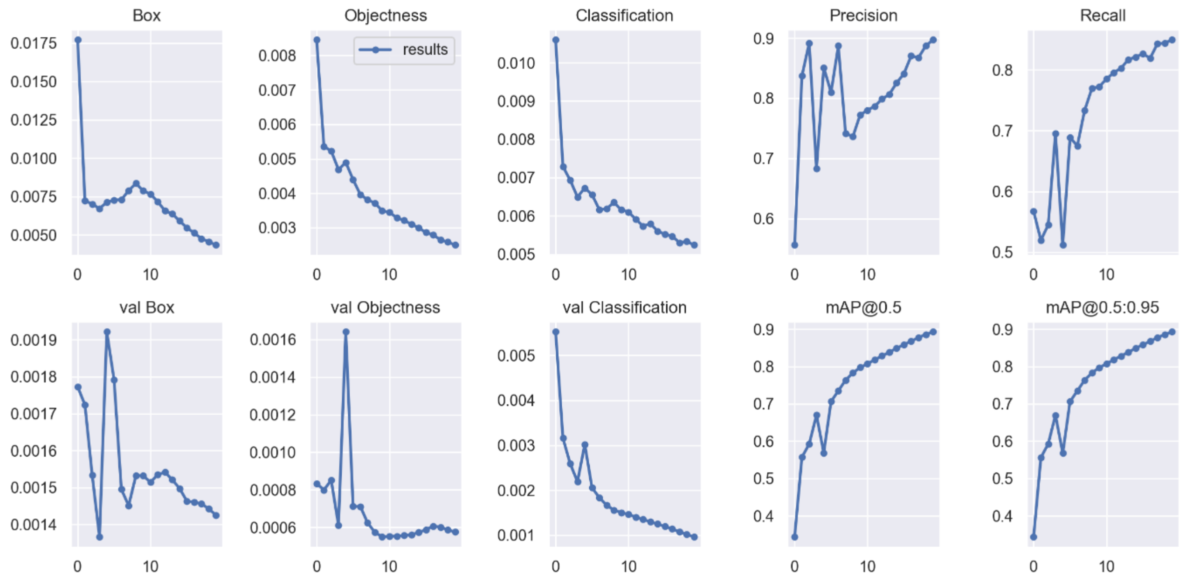

Figure 2: Results of model, noting Precision and mAP scores

The Confidence vs Precision score of model for all 6 classes (figure 3):

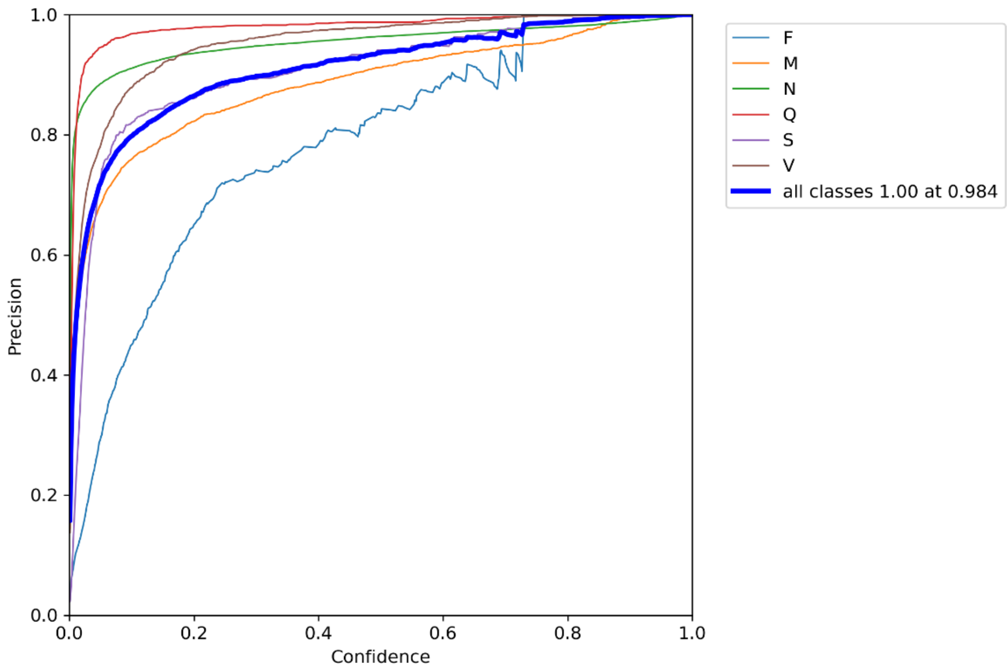

Figure 3: Precision Score diagram

Figure 4 and 5 compares test sample and test output of the ecg images:

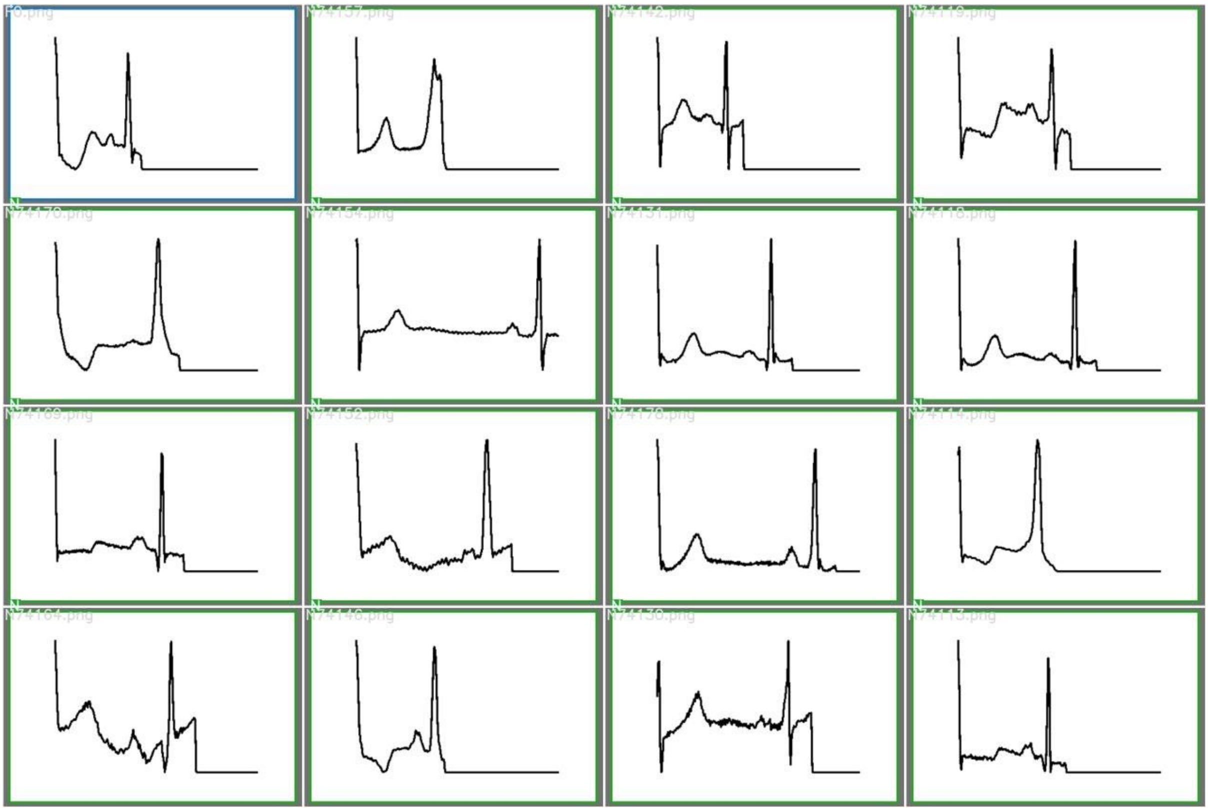

Figure 4: example of test sample

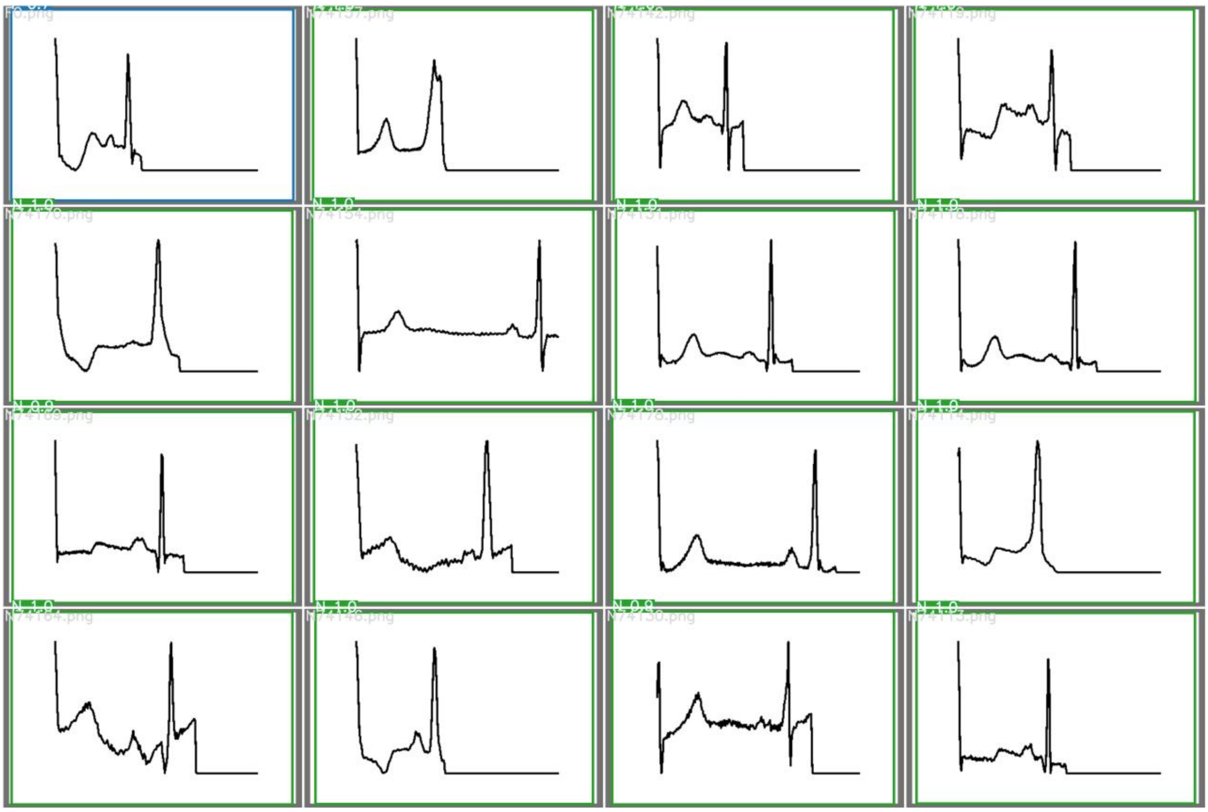

Figure 5: example of test output

Confusion matrix:

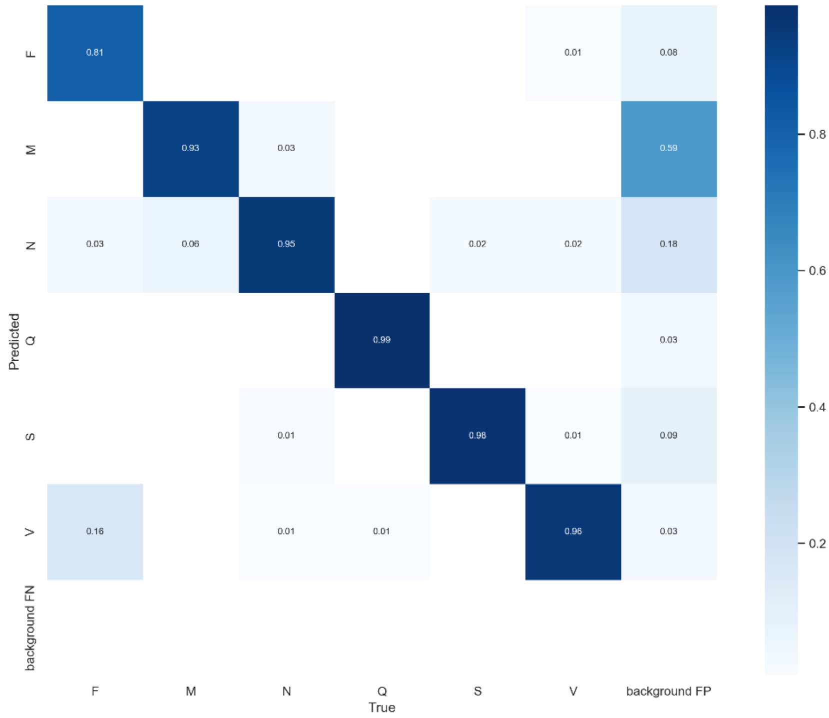

Figure 6: confusion matrix for yolov7

Another pretrained model VGG19, using Adam optimizer and categorical_crossentropy loss function, training 99199 images in batch sizes of 32 images at a time, and using a validation set of 24799 also in batch sizes of 32 images at a time for 10 epochs. The result is a validation accuracy of 0.9998 which performs extremely well.

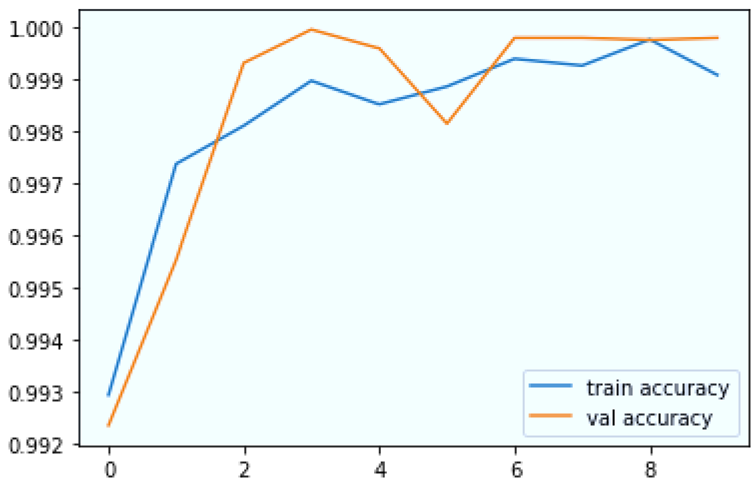

Figure 7: accuracy for vgg19 model

As such, we believe that indeed convolutional neural networks do perform well in training a model to classify ECG images for triage.

Future improvements:

As of now there is no dataset available that combines both patients' details and symptoms as well as images of their scans such as ECG, which can be used to train a model to complete a full triage process. However, we believe that if such datasets are available and well labelled, existing deep neural networks would be able to perform well in training these models to classify patients for triage.