Disease Diagnosis using Deep learning

Model Training

- Unlike machine learning, deep learning automatically extract useful features from the data.
- Framework for deep learning
  - MXNet (AWS)
  - Pytorh (facebook)
  - Tensorflow (Google)
- Each framework has easy to use wrapper
  - Autogluon (MxNet)
  - Pytorch Lightening/ FAST.ai
  - Keras (TensorFlow)

Model Training

- Currently, I have developed the pipeline for all sort of classification task . This will save our time as we have prebuilt pipelines ready. Goal is to pass pre-trained models and our data to these pipeline and we will get new model as output.
- Next, I have to add clinical features to these pipeline such as doctors notes and other tabular data such as demography (not completed yet).

Model Inference

- Memory Issues
  - MxNet produced best result as it do hyperparameters tuning, but it size is very large. Each time a user sent a query for prediction, the model will load in memory and if multiple models are loaded, there will be memory error.
  - With change in version of the package, the old models will stop working, and we have to maintain the different versions of framework. This also cause memory issue
  - If different models are created in different frameworks, we have to keep different framework in our code.
- Solution
  - We will transfer all of our model to Open Neural Network Exchange ecosystem, which will resolve versioning and framework issues.

Open Neural Network Exchange (ONNX)

- ONNX make the machine learning and deep learning models independent of programing language , framework and hardware.
- We train our model on float16, (old GPUs did not support this), this reduce the training time and model size significantly. But for inference, we can use any GPU or CPU

Dataset used

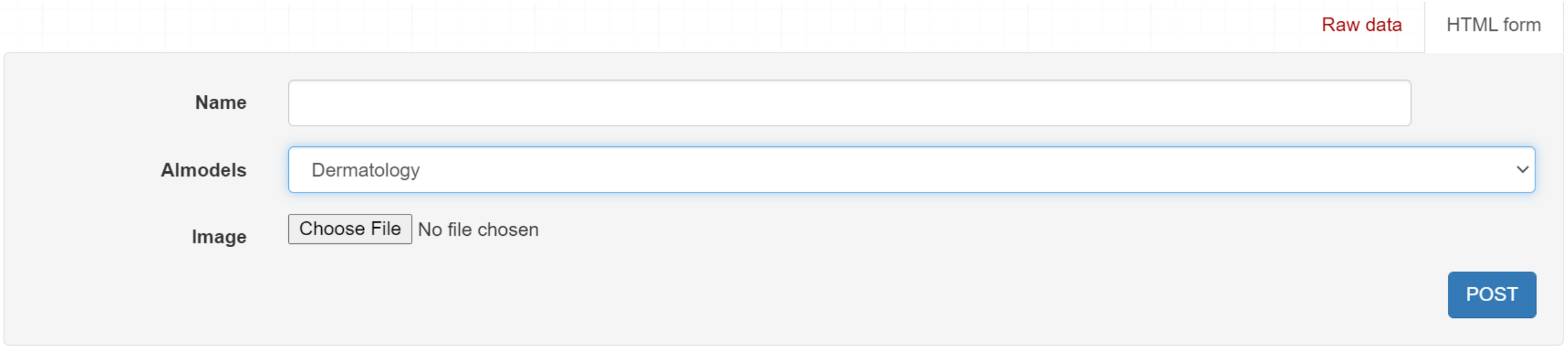
- ChexPert is chest radiograph dataset composed of 0.25 million radiographs of 65,000 patients, labelled by 3 radiologists. It can classify 14 diseases.
- ECG data is collected in Pakistan, and it has 2000 reports of patients.
- Skin dataset composed of 3000 images captured via mobile camera for 6 skin diseases.
- Covid, an online Xray model picked for our task.

Future work

- Create pre-train models of any medical data available.
- If we want to use the models in production, we have to further reduced the model size. Currently a single model size approximately 30 mb and hopefully we have multiple models to get better performance. We will do quantization to reduce model size.
- For training, we need some GPU/TPU resources, the cheapest solution is google Colab, which cost 10$/mo, but currently not available in Pakistan. This will help to get pre-trained models of data available on net before we actually start working on our data.

Conclusion

- Created a pipeline for all sort of classification task
- Created a django rest framework API
- Need to create a pipeline for localization task



{"user": "Talha", "model": "covid", "prediction": "positive"}