Building Machine Learning Models to perform COVID-19 Diagnosis from chest X-ray images

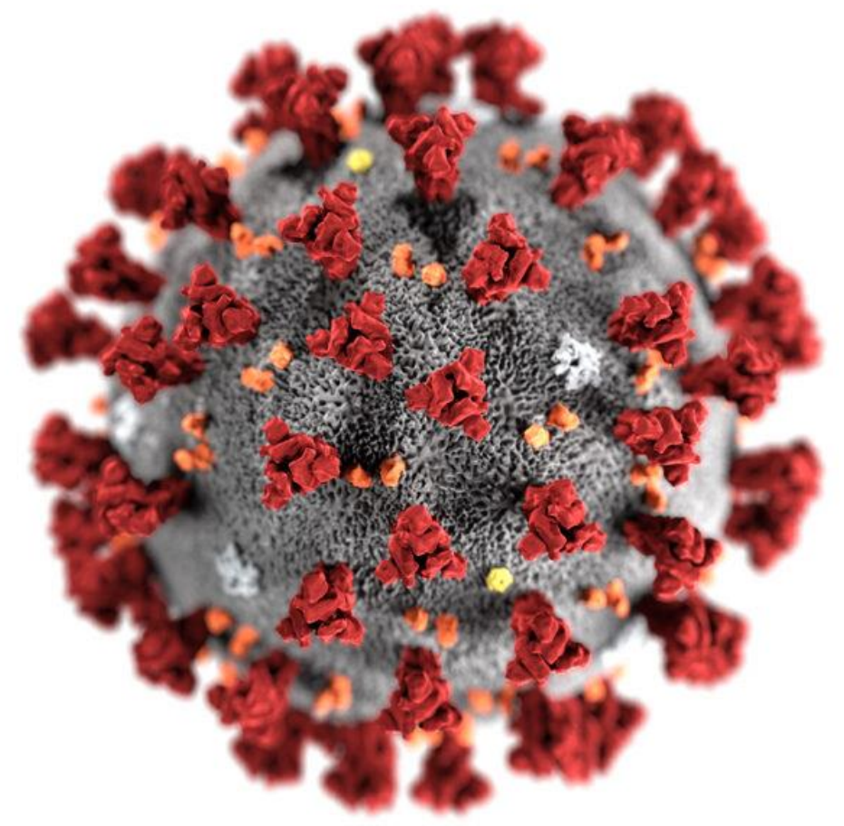
Texas A&M
Department of Computer Science and Engineering

Aaryan Kothapalli, Gabriel Stella, Keishla D. Ortiz-Lopez, Qusai Amer, Yiqing Zhao

Professor: Dr. Theodora Chaspari

Problem Statement

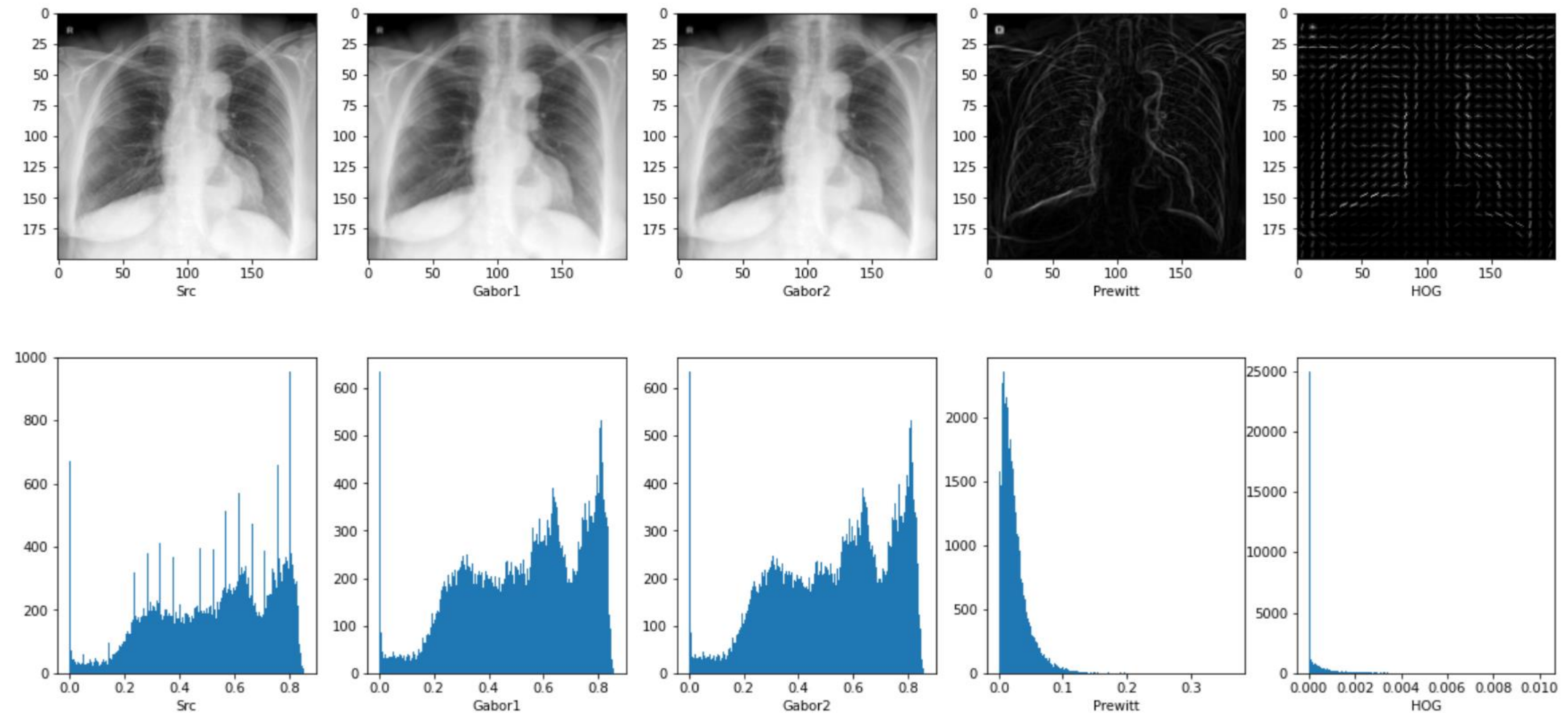
COVID-19 is an ongoing pandemic that is caused by SARS-CoV-2. The current popular way to detect the virus is by an antibodies test or temperature test. To widen the ability to detect the virus in patients, we use Machine Learning to diagnose the virus using patient X-ray images.



Method

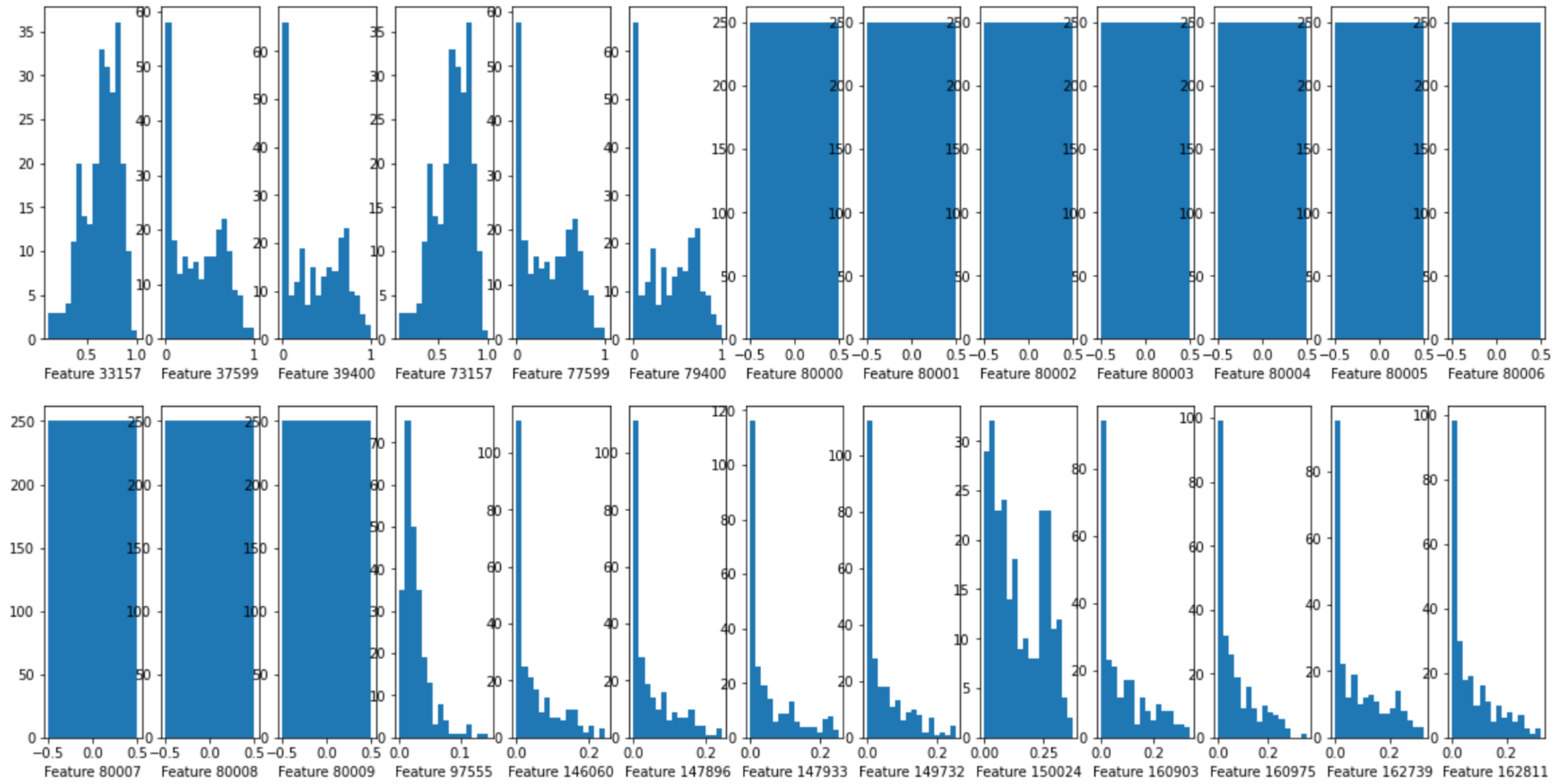
- **Libraries Used:** Numpy, Pandas, Pillow, Matplotlib, and Scikit-learn
- **Preprocessing:**
  - Images are resized to 200x200 for feature extraction, visualization, scoring, and selection process.
  - Images are further reduced to 100x100 for CNN to improving performance and shave off processing time.
- **Feature Extraction:**
  - Two Gabor filters, Prewitt edge filter, and HOG are used.
  - Feature vector generated by concatenating all features.
- **Feature Visualization:**

Left to right: source, Gabor1, Gabor2, Prewitt, HOG



Method (contd.)

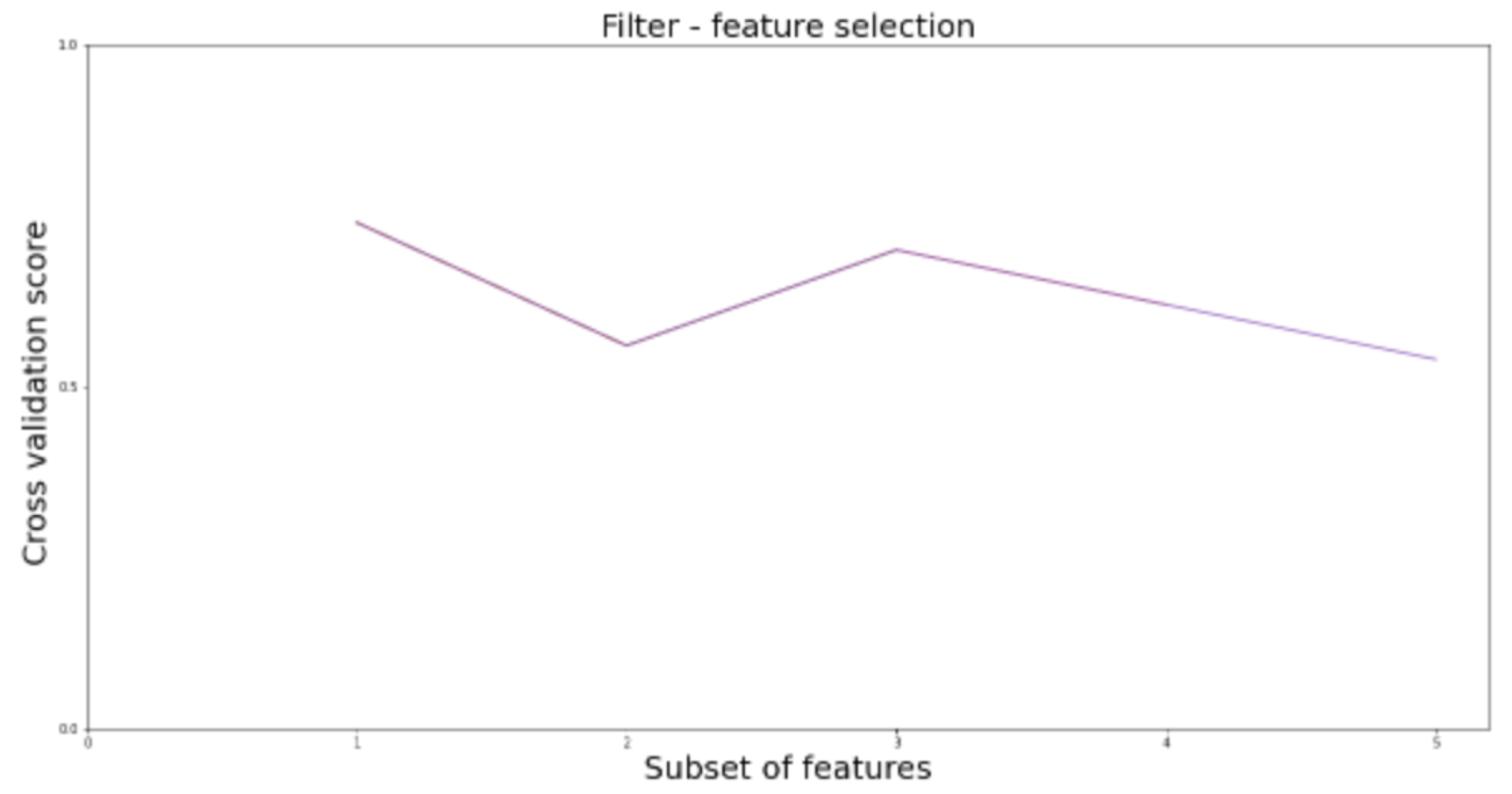
- **Feature Scoring:**
  - 3 types of scores are generated: Fisher score, Gini index, Conditional entropy.
  - For each score type, features are ranked and the top ten are collected.
  - The 10 best features from each group are combined and displayed as a histogram of 30 features. Duplicates are removed from the chosen features to prevent overlapping. After duplicate detection, 26 best features are displayed.



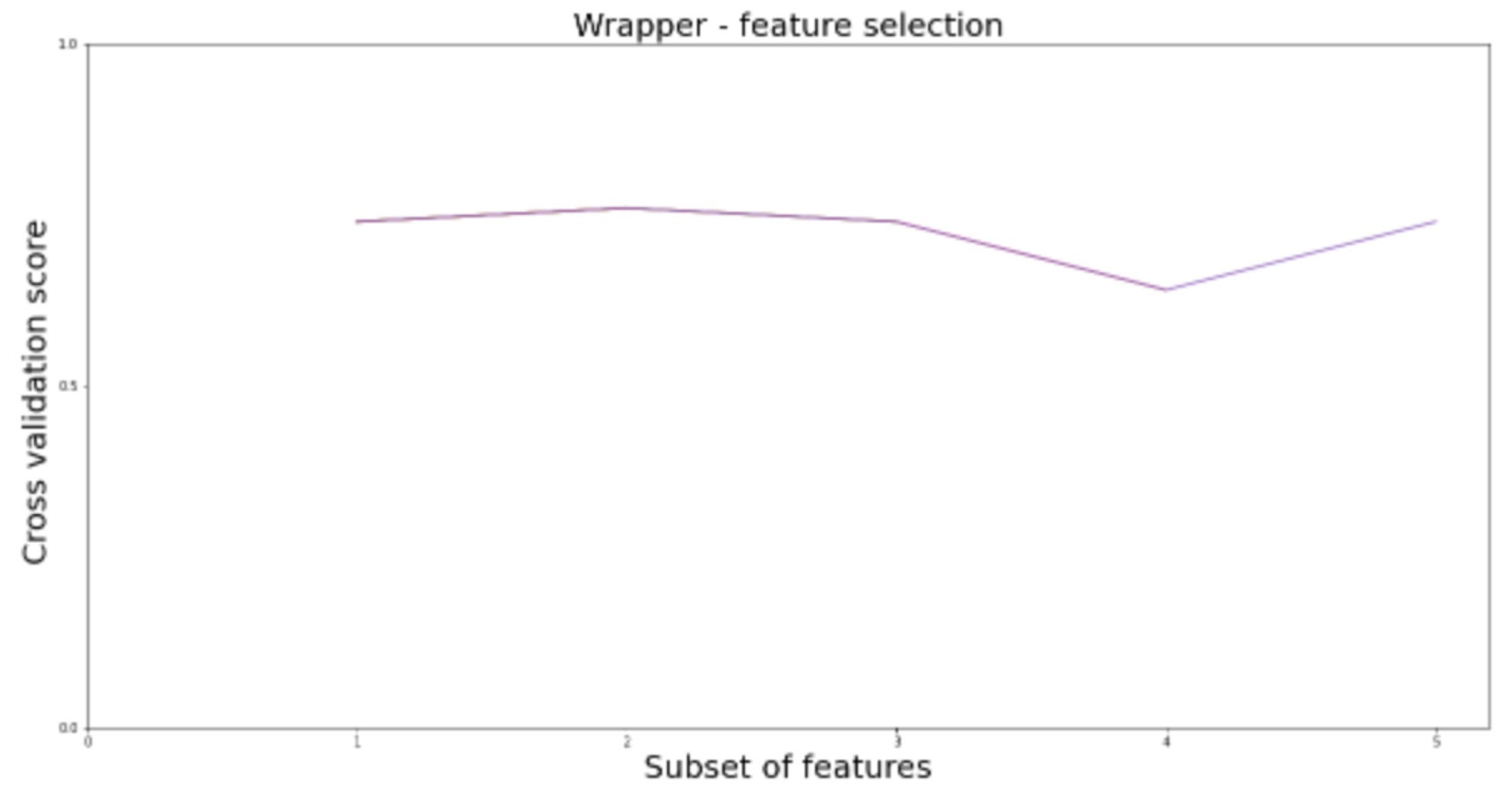
- **Feature Selection:**
  - About 162,800 potential features were produced.
  - Filter method: each feature ranked based on univariate metrics. SelectKbest from scikit-learn is used.
  - Wrapper method: Recursive Feature Elimination is used to select best features. Worst performing features are removed until only the best features remain. Wrapper method was chosen because of its higher accuracy coverage.
  - Both methods were cross-validated in a 5-fold split.
- **Improving Performance:**
  - Adaboost was used with 5-fold cross validation split.
  - To further improve performance beyond 0.7, a CNN model with Hyperopt was chosen to find the best kernel size, stride, dropout, no. of convolutional and max pooling layers, and activation layers within 3 dense layers.

Results

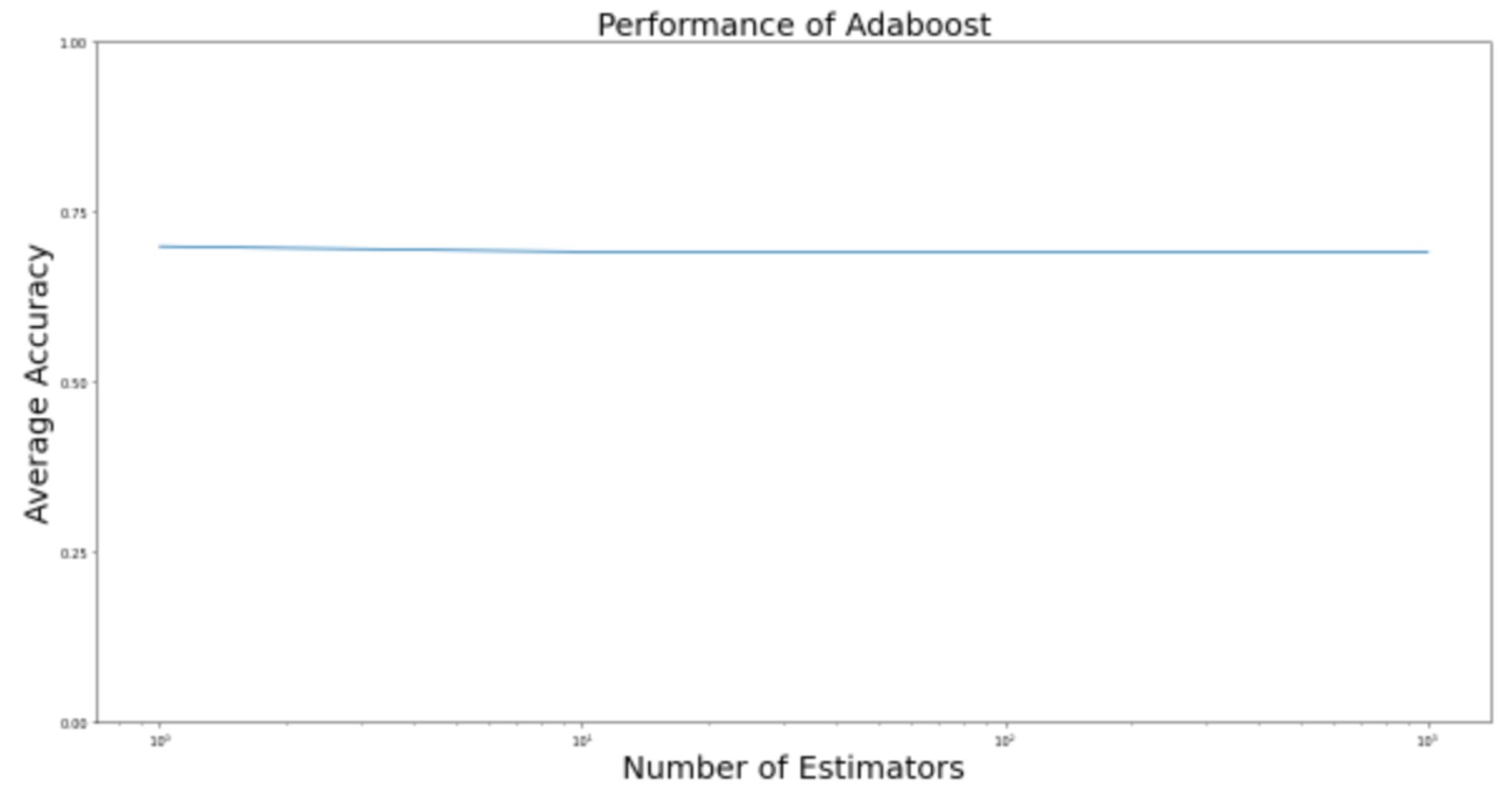
Filter Method, Average Accuracy: 0.632



Wrapper Method, Average Accuracy: 0.724



Adaboost, Average Accuracy: 0.7



CNN Model with HyperOpt:

Accuracy after HyperOpt	
Convolutional Layers	2
Max Pooling Layers	1
Dense Layers	3
Activation Function	Rectified Linear (ReLU)
Kernel Size	3
Dropout Probability	0.22
Resultant Accuracy	**92.40%**

Acknowledgment

Our team did not infringe on any copyright in the making of this project. We used open-source libraries for all code that was implemented. Everything was obtained legally and ethically.