TASK DESCRIPTION

The DFDC Challenge is focused on developing effective techniques for detecting deepfake videos. Deepfake videos are created using artificial intelligence and machine learning algorithms to manipulate and replace the original content of a video with fabricated content, often resulting in highly convincing and realistic fake videos. The challenge involves building a model that can accurately detect deepfake videos from a given dataset, which consists of a large number of real and fake videos.

INPUTS

The inputs to my Simple RNN model consists of video frames which has been preprocessed.

EXPECTED OUTPUT

The expected outputis a binary classification indicating whether a given video is Fake or Real. The model should produce a probabaility of confidence score that represents the likelihood of a video being a DeepFake. A threshold can be set to convert the probability to a binary decision. If the probability is above a certain threshold, the video is classified as DeepFake ,otherwise as Real.

METHOD USED: SimpleRNN

POTENTIAL CHALLENGES EXPECTED

1)VARIABILITY IN DEEPFAKE TECHNIQUES: DeepFake techniques are constantly evolving and new techniques may emerge over time. My model may not be effective in detecting new or unseen DeepFake techniques that were not present in the training data.

2)COMPUTATIONAL COMPLEXITY: Training deep learning models such as RNN can be computationally expensive and time consuming. Managing computational resources efficiently and optimizing the model training process may be challenging

LIMITATIONS OF THE SIMPLE RNN MODEL

1. Short-Term Memory: Simple RNNs suffer from the vanishing gradient problem, which can result in poor performance in capturing long-range dependencies in sequences. They have limited memory and struggle to capture information from sequences that are too long, resulting in reduced performance in tasks that require modeling long-term dependencies.
2. Gradient Vanishing and Exploding: Simple RNNs are prone to the issue of gradient vanishing or exploding during backpropagation. This can lead to

difficulties in training the model effectively and may require careful initialization, regularization, or other techniques to mitigate the problem.

3. Lack of Explicit Memory Mechanisms: Simple RNNs do not have explicit memory mechanisms, such as memory cells or gates, like more advanced recurrent architectures such as LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit). This can limit their ability to effectively capture complex patterns and dependencies in sequences.

PRECISION: 0.3

F1: 0.57

ROC AUC SCORE: 1