1. PROBLEM DOMAIN

Fake News Detector Using Naive Bayes Algorithm Problem Domain

Introduction

Abstract: In the era of digital text where false information propagates like never before, fake news detection has emerged as an important problem. Fake news is a term used to describe false information presented as news, often to mislead or manipulate the audience for political or social ends. A Fake News Detector uses methods of computation to distinguish a news article as being real or fake.

Problem Statement

The problem involves building a system capable of classifying news video as *Fake or Real* based on their content. This classification can assist individuals, organizations, and governments in reducing the impact of misinformation.

Objectives

- Automate the classification of news a video.
- Improve the accuracy of fake news detection using a Naive Bayes algorithm.
- Provide a scalable and efficient solution to handle a large volume of online news video.

Solution Overview: Naive Bayes Algorithm

The Naive Bayes algorithm is a probabilistic classification method based on Bayes' Theorem, assuming feature independence. It's highly efficient and works well for text-based classification tasks, such as fake news detection.

Key Features of Naive Bayes Algorithm

1. **Simplicity**: Easy to implement and computationally efficient.
2. **Probabilistic Approach**: Calculates the probability of a news article being real or fake.
3. **Text Classification**: Suitable for bag-of-words models and natural language processing tasks.

Steps to Implement Fake News Detection

1. **Data Collection**
   - Collect labeled datasets of fake and real news articles (e.g., Kaggle Fake News Dataset, BuzzFeed News articles).
   - The dataset includes features like article text, title, and publication date.
2. **Data Preprocessing**
   - Tokenization: Break down text into individual words.
   - Remove stop words: Eliminate common words (e.g., "is," "the") that don't add meaning.

   - Stemming/Lemmatization: Reduce words to their root forms.
   - Vectorization: Convert textual data into numerical form using techniques like TF-IDF or CountVectorizer.
3. **Feature Selection**
   - Select important features like word frequencies, bigrams, or term importance.
   - Reduce dimensionality to retain meaningful information.
4. **Training the Naive Bayes Model**
   - Apply Naive Bayes to the preprocessed data.
   - Types of Naive Bayes models:
     - **Multinomial Naive Bayes**: For word frequency features.
     - **Bernoulli Naive Bayes**: For binary occurrence features.
5. **Model Evaluation**
   - Evaluate the model using metrics like accuracy, precision, recall, and F1-score.
   - Use k-fold cross-validation to assess robustness.
6. **Deployment**
   - Integrate the model into an application or API.
   - Allow users to input news articles for real-time classification.

2. PROPOSED TREATMENT

1. Data Collection

- Collect labeled fake and real news videos from trusted sources.
- Include diverse video formats and languages for better model generalization.

2. Data Preprocessing

- **Audio**: Extract speech and convert it to text using ASR tools (e.g., Google Speech-to-Text).
- **Text**: Clean transcribed text using NLP techniques (e.g., tokenization, stop-word removal).
- **Visual**: Extract video frames to analyze facial expressions, background, and detect tampering using tools like OpenCV.

3. Feature Extraction

- **Text**: Use methods like TF-IDF or embeddings (e.g., BERT).
- **Audio**: Analyze tone and emotional cues with tools like librosa.
- **Visual**: Detect anomalies or deepfake elements with CNN-based models.

4. Model Development

- Combine features from text, audio, and visual components in a multimodal model.
- Use Naive Bayes for text and advanced CNNs or RNNs for video and audio features.

5. Deployment

- Build a system where users can upload videos for real-time classification as fake or real.
- Ensure scalability and periodic retraining with updated datasets.

3. PLAN OF WORK

Phase 1: Research and Planning

- Conduct research on fake news detection methods.
- Select tools, frameworks, and datasets.
- Define scope, objectives, and team roles.

Phase 2: Data Collection and Preparation

- Collect fake and real news videos from trusted sources.
- Annotate data and preprocess:
    - **Audio**: Convert speech to text using ASR.
    - **Text**: Clean and preprocess text.
    - **Visual**: Extract frames and detect anomalies.

Phase 3: Feature Engineering

- Extract features:
    - **Text**: Use TF-IDF or embeddings.
    - **Audio**: Analyze tone and emotion.
    - **Visual**: Detect deepfake or tampering via CNNs.

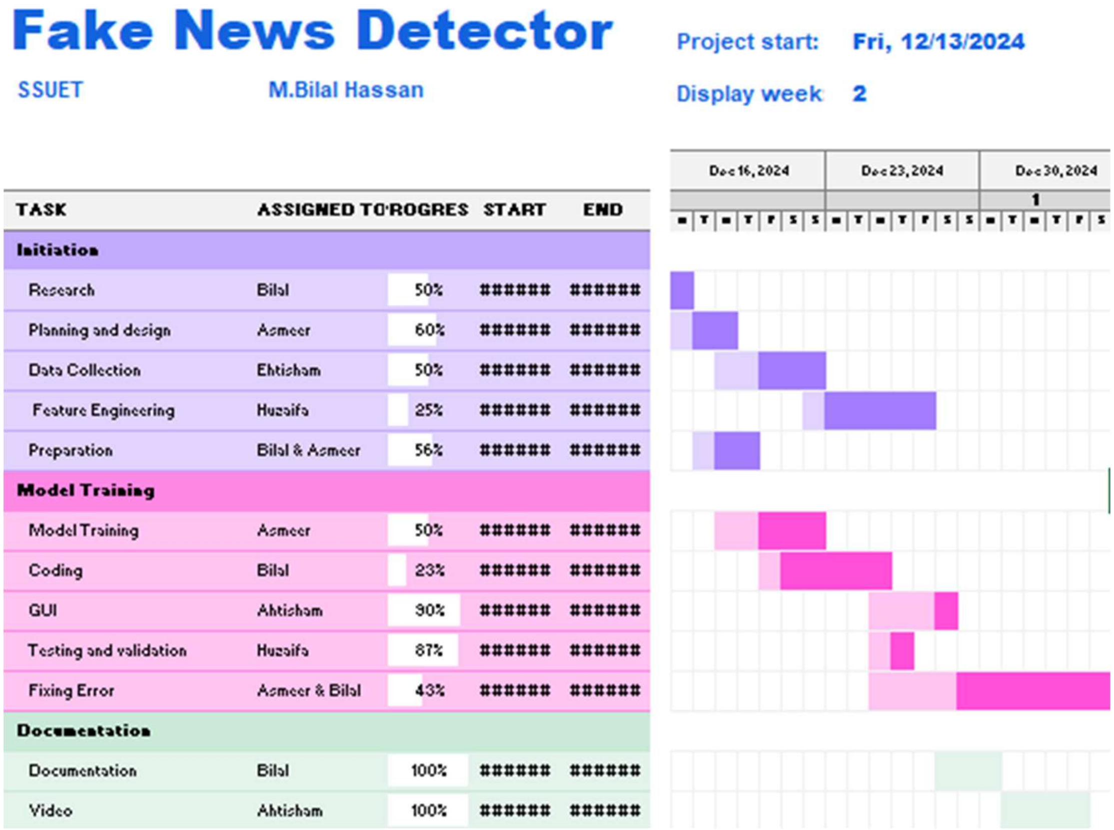
Phase 4: Model Development

- Build models for text, audio, and visuals.
- Integrate into a multimodal model for fake news classification.

Phase 5: Testing and Deployment

- Test the model for accuracy and scalability.
- Develop a user-friendly system for real-time video classification.

4. PROJECT SCHEDULING



Fake News Detector

SSUET M.Bilal Hassan

Project start: Fri, 12/13/2024

Display week 2

TASK	ASSIGNED TO	PROGRES	START	END
Initiation				
Research	Bilal	50%	######	######
Planning and design	Asmeer	60%	######	######
Data Collection	Ehtisham	50%	######	######
Feature Engineering	Huzaifa	25%	######	######
Preparation	Bilal & Asmeer	56%	######	######
Model Training				
Model Training	Asmeer	50%	######	######
Coding	Bilal	23%	######	######
GUI	Ahtisham	90%	######	######
Testing and validation	Huzaifa	87%	######	######
Fixing Error	Asmeer & Bilal	43%	######	######
Documentation				
Documentation	Bilal	100%	######	######
Video	Ahtisham	100%	######	######

5. SOFTWARE AND HARDWARE SPECIFICATIONS

Hardware Requirements

- **Processor**: Minimum 2 GHz; Recommended 3 GHz or more (multi-core).
- **Memory (RAM)**: Minimum 4 GB; Recommended 8 GB or more.
- **Storage**: Minimum 32 GB; Recommended 64 GB or more (SSD preferred).
- **GPU**: NO GPU needed
- **Sound Card**: Required for audio extraction and analysis.
- **Camera & Microphone**: No need

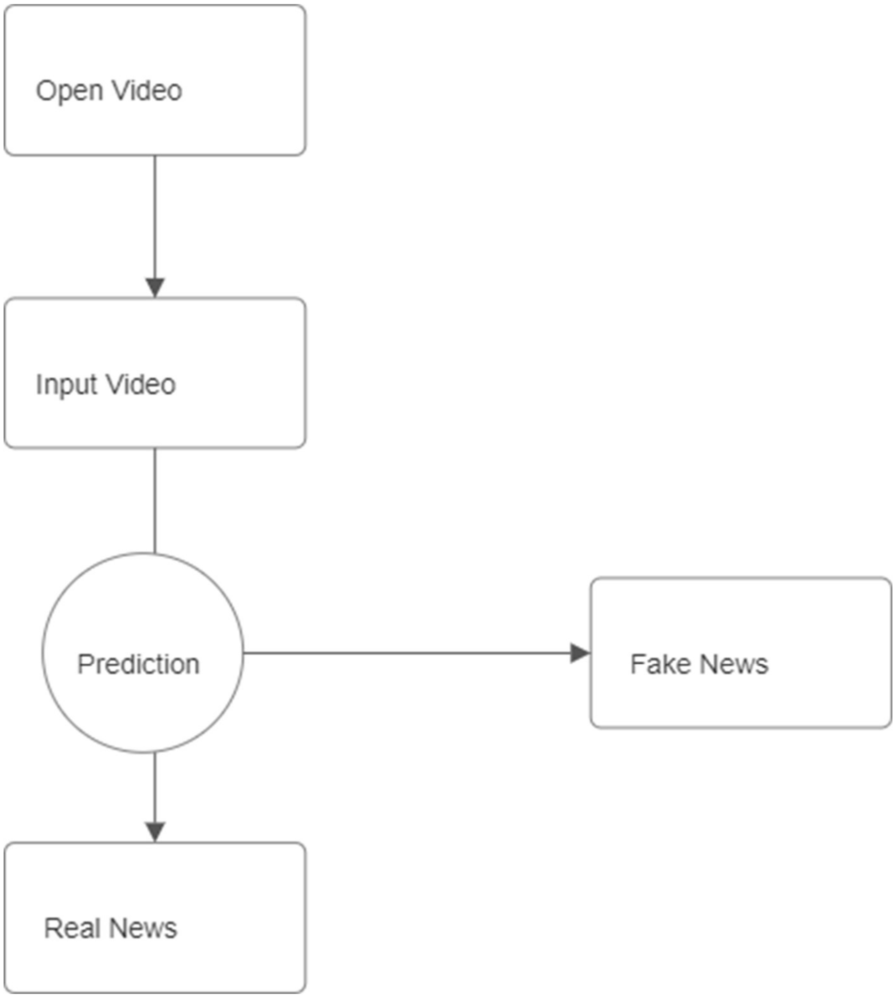
Software Requirements

- **Operating System**: Windows 10/11, macOS, or Linux (Ubuntu 20.04+).
- **Programming Tools**: Python 3.9+ with libraries (e.g., TensorFlow, PyTorch, Jupiter, librosa, scikit-learn).
- **Database**: Excel
- **Video Processing Tools**: No Need.
- **Speech-to-Text Tools**: Whisper or Google Speech-to-Text API.

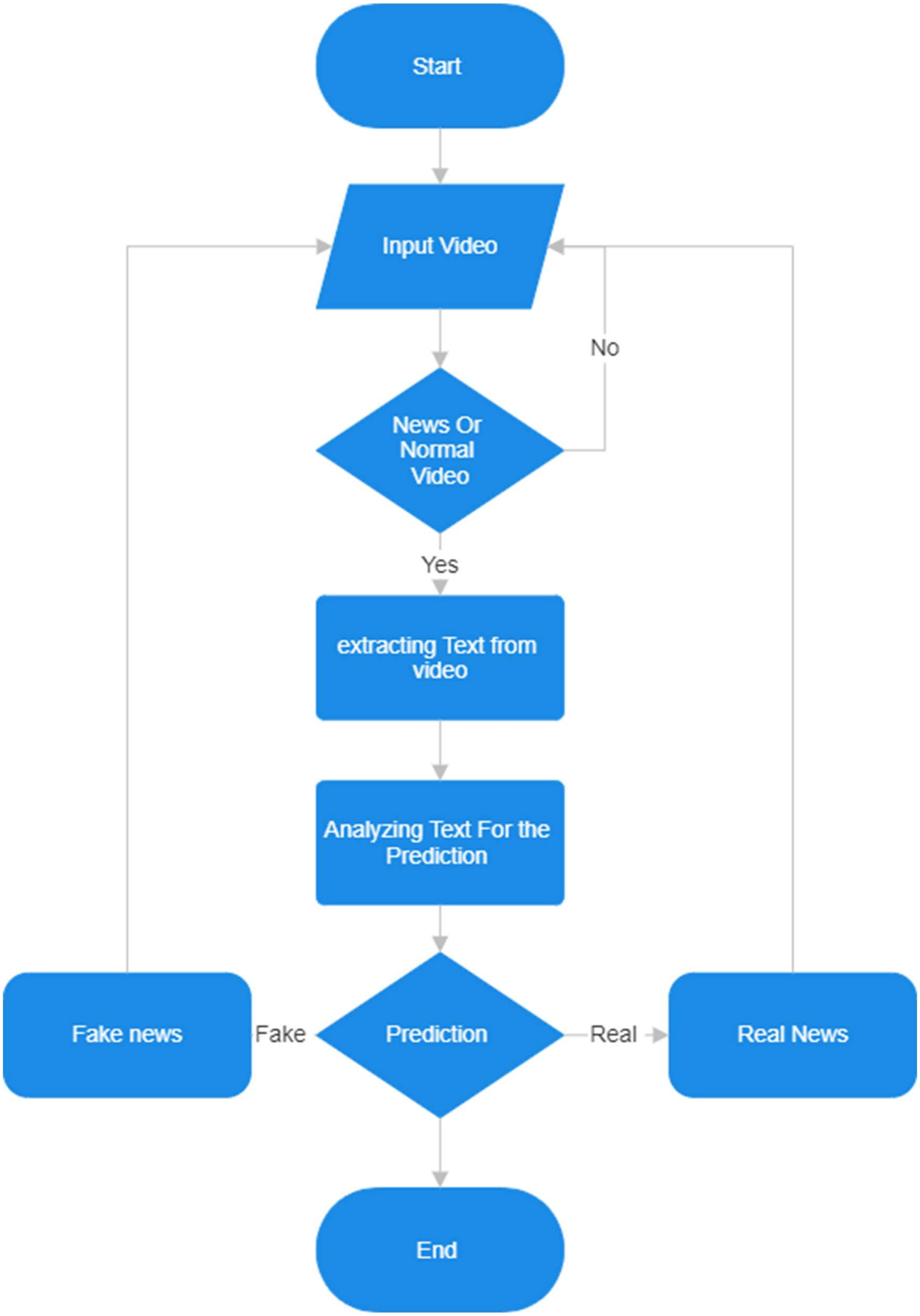
- **Development Environment**: Jupyter Notebook, VS Code, or PyCharm.

This setup ensures seamless video analysis for detecting fake news using text, audio, and visual features.

6. BLOCK DIAGRAM



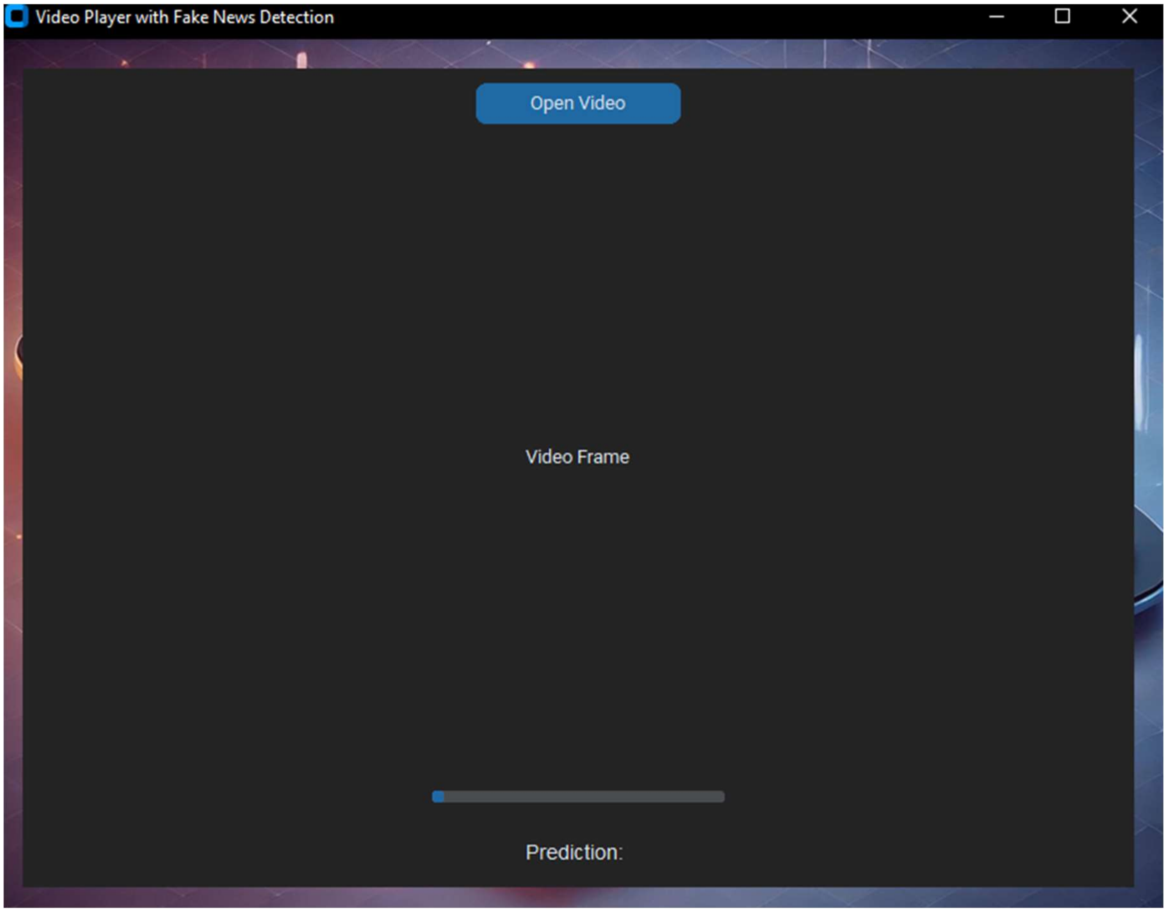
7. SYSTEM FLOW DIAGRAM



8. USER GUIDE

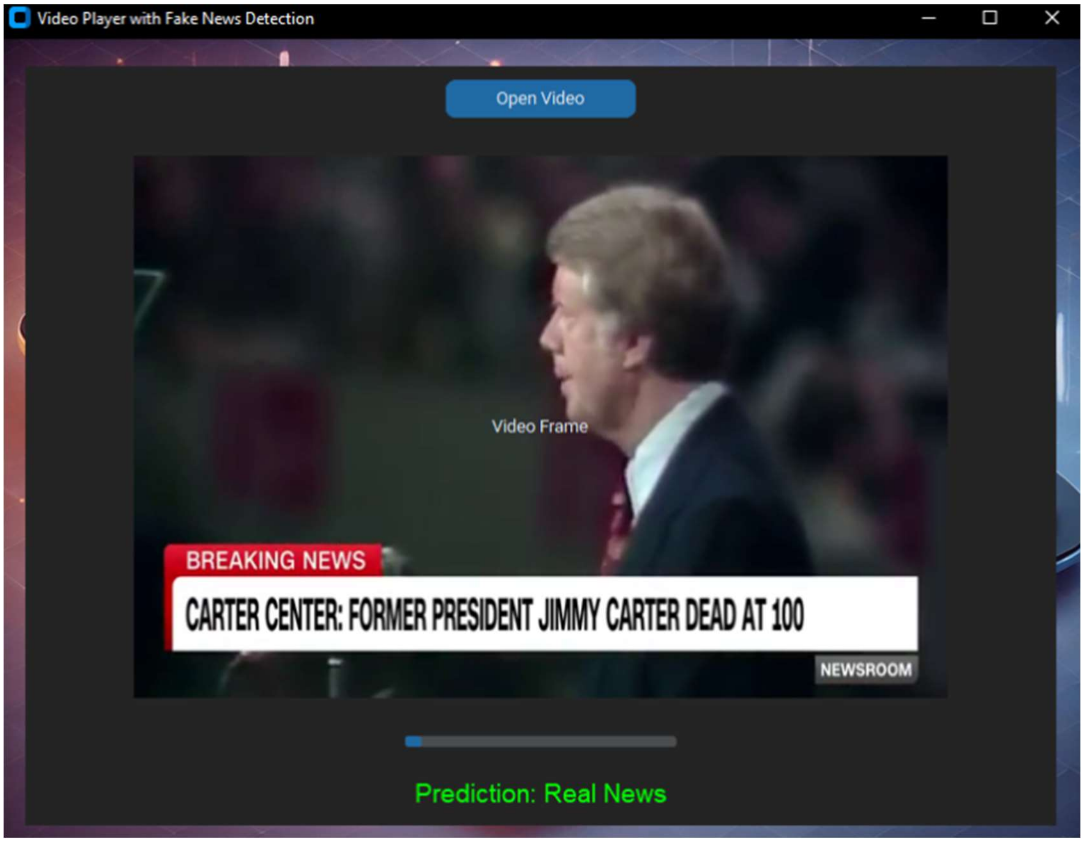
1.Main Interface Of Fake News Detector By Video:

The primary interface of the fake news detector is designed to be user-friendly and straightforward. Simply click on the "Open Video" button to upload a video from your device.



2. How App Predict The Scenario

The app analyzes the scenario by extracting text from the video, and the prediction—indicating whether the content is real or fake—will appear at the bottom center of the video.


Video Player with Fake News Detection
Open Video
Video Frame
BREAKING NEWS
CARTER CENTER: FORMER PRESIDENT JIMMY CARTER DEAD AT 100
NEWSROOM

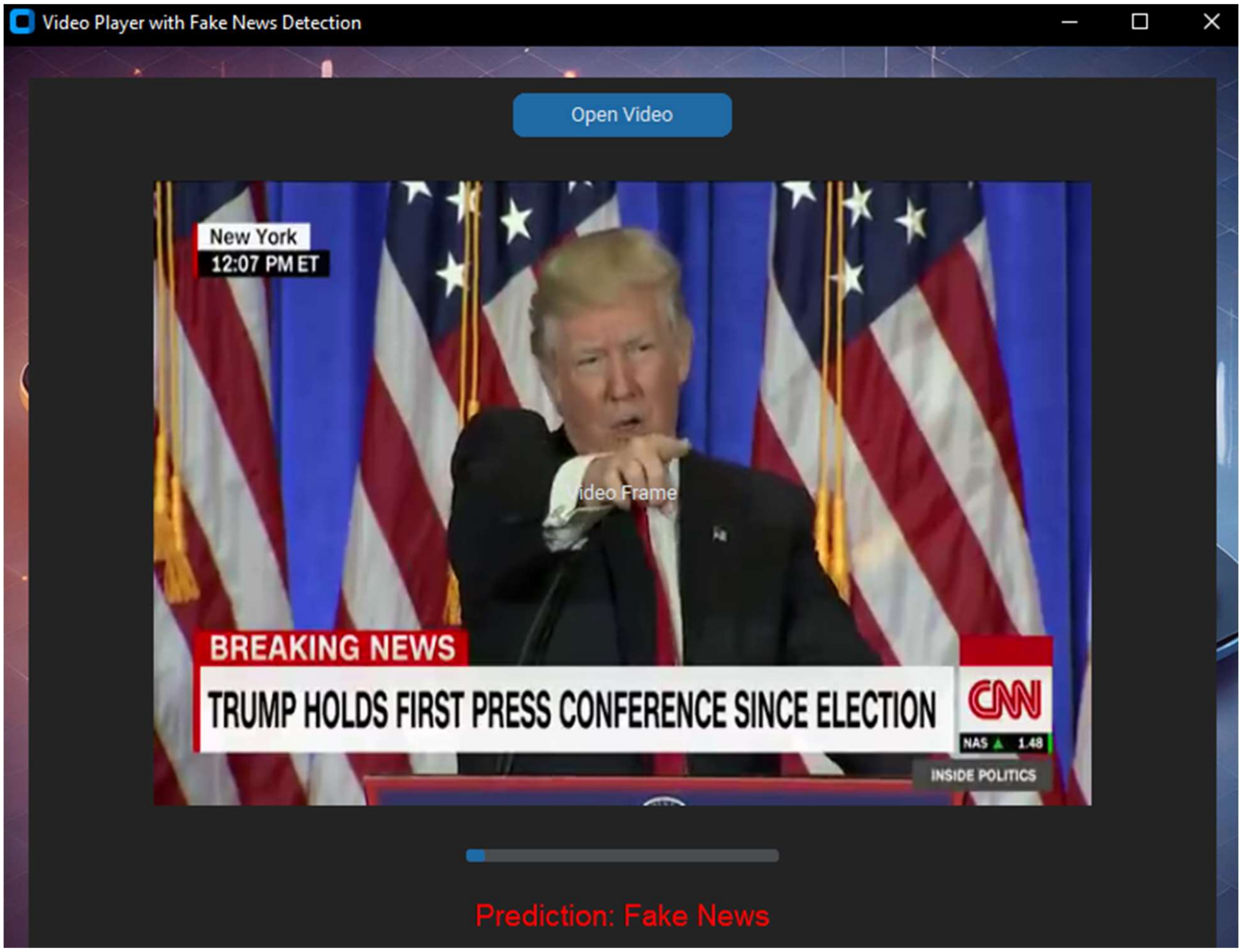
Prediction: Real News

3. Fake Video Prediction:

Fake news prediction involves analyzing the content of a video by extracting text from it and determining its authenticity. The process typically works as follows:

1. **Text Extraction**: The system uses advanced techniques like Optical Character Recognition (OCR) to extract visible text from the video frames. This may include subtitles, headlines, or embedded textual elements within the video.
2. **Content Analysis**: The extracted text is then analyzed using a database of verified information, linguistic models, and machine learning algorithms. These tools assess the credibility of the text based on factors such as language patterns, sources, and the presence of sensational or misleading content.
3. **Prediction Outcome**: Based on the analysis, the app predicts whether the video contains fake or real information. This result is displayed prominently, such as at the bottom center of the video, ensuring clarity for the user.

This approach combines artificial intelligence and data verification to help users identify and combat the spread of fake news effectively.


Video Player with Fake News Detection
Open Video
New York
12:07 PM ET
Video Frame
BREAKING NEWS
TRUMP HOLDS FIRST PRESS CONFERENCE SINCE ELECTION
CNN
NAS ▲ 1.48
INSIDE POLITICS

Prediction: Fake News

REFERENCES

Here's a concise list of references for fake news detection:

Research Papers

1. **"Fake News Detection on Social Media: A Data Mining Perspective"** - Shu et al.
2. **"Automatic Fake News Detection: Are We There Yet?"** - Zubiaga et al.

Tools and Libraries

- **Tesseract OCR**: Extracts text from images or video frames.

- **InVID Verification Plugin**: Analyzes video content for authenticity.

Fact-Checking Platforms

- Snopes
- PolitiFact
- FactCheck.org

Datasets

- LIAR Dataset
- Fake and Real News Dataset

Courses

- Coursera: AI for Everyone.
- Udemy: NLP with Python.