Detailed Version:

1. The model uses preceding 30 frames for training procedure on the videos where the number of frames are generated using the opencv library function called:

```
self.cap=cv2.VideoCapture("video.mp4")
## provide the total number of frames of given video from given code
self.frame_total = self.cap.get(cv2.CAP_PROP_FRAME_COUNT)
## then the given function is used to read each frame sequentially
ret, image = self.cap.read()
```

2. Given video having frame rate 30 fps, we get output labels every 1 second without overlapping previous segment(size: 30 frame) from given model. In case, we need output labels every 0.5 seconds, there must be a overlapping of 15 frames between previous video segment and current segment.
3. Similarly , if we have sequence of frames suppose 60 frames for video with frame rate 30 fps, the model outputs labels 2 times only.
4. The output labels is based on three specific detection tasks which takes the given 30 sequential frames as inputs.
    a. Action Units Detection:
        i. Consists of 12 output labels in form of [0, 0, 0, 0, 0, 0, 0, 0, 0, 0,1,1]. Each index in given multi-class encoding refers to following action units: {Inner brow raiser, outer brow raiser, Brow lower, Upper Lid raiser, Cheek raiser, Nose raiser, Lip corner puller, Lip corner depressor, Chin raiser, Lip Stretcher, Lips part, Jaw drop}.
        ii. Resnet50 is utilized to each frame to output action labels which is averaged for all the 30 frames and thus given final labels is outputed.
        iii. Also the corresponding feature dimension of 2048 from each frame is also averaged for deception detection tasks.
    b. Emotion units Detection:
        i. Consists of 7 output labels in form of [0.2016, 0.1439, 0.3892, -0.5354, 0.3730, -0.6479, -0.8379] where argmax output the respective emotion unit which is integer value range from 0 to 6. Each index refers to following emotion units: {Happy, Angry, Disgust, Fear, Sad, Neutral, Surprise}.
        ii. Emotion-FAN code is utilized for 30 frames to output single 1024 dim feature vector using CNN+ relation-attention from paper which is concatenated with above Action features for deception detection.
        iii. It also outputs 7 output labels using final softmax activation function.
    c. Deception Detection:

i. Outputs single value as either deceptive (1) or Truth (0)
ii. Finally the concatenated Action Units features and Emotion Units features are passed through SVM for binary classification tasks.

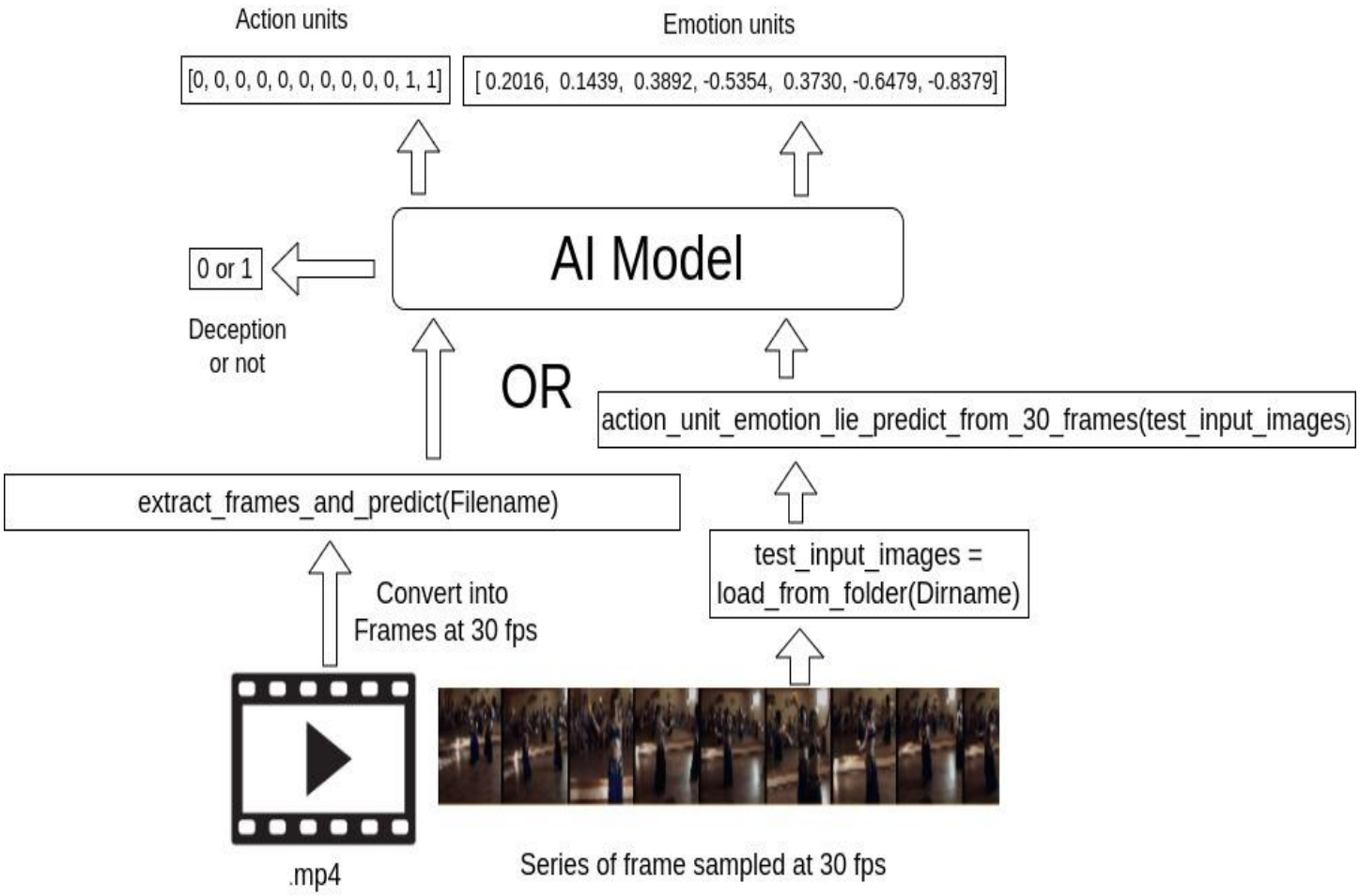

Fig: Simple diagram of whole process

Python_API()

1. So, We have created the simple python api functions which can take both videos and sequences of frames as the input and output the labels for respective different recognition tasks.
2. The main_file called lie_GUI.py from which the main scripts runs, it is replaced with the another file lie.py .It can process both video and sequences of frames which is limited in original file and also we removed the pyqt functionality so that it can be integrated into our applications.
3. Functioning of input functions:
   - test_input_images = load_from_folder(Dirname): For testing purpose, we passed “Dirname” as the argument which consists of sequences of images.
   - outputs= action_unit_emotion_lie_predict_from_30_frames(test_input_images): The outputs is the tuple(aUs, emotions, lie) i.e tuple of Action units labels + Emotion Units labels + lie label where each units labels are used for different predictions tasks as specified for only 30 frames for now.
   - mylist_of_timestamped_outputs = extract_frames_and_predict(Filename): It outputs the list of tuple of predictions for each 30 frames across the videos until end. It is in format as given by:

     ```
     List [
        (AUs_1, emotions_1, lie_1),  # output in sec 1
        (AUs_2, emotions_2, lie_2),  # output in sec 2
        (AUs_3, emotions_3, lie_3),  # output in sec 3
        ...
      ]
     ```

     Other useful function used inside the library:

   - api= lie(): class instantiate all the input and output labels as needed.
   - A_U,E_U,D=api.predictions(list_images): function passes the api object along with the sequences of 30 images matrix. It process the 30 images and extract action_embeddings + emotion_embeddings which are fed to pred() function of show() class to output the tuple of actions units, emotion units, deceptive or not

- A_U,A_E=AU_pred.run() or Action_class._pred(image,config) function predicts the labels for 12 actions units and corresponding feature dimension.
- E_U,S_E,R_E=Emotion_class.validate(input) takes the 30 frames as input and output 7 emotion labels with fully connected layer,self_embedding and relation_embeddings.
- D=SVM_model.predict(input): takes the concatenated features of relation embeddings and global_action_unit embeddings(A_E) and output label of value 1 or 0 for deception detection.