**Gesture based Misbehaviour Detection in Online Exam**

**Business view of Project:**

Nowadays, more and more online examinations are holden in Colleges and Institution, which calls for the deployment of examination surveillance systems. Various related techniques have been proposed for this purpose, for example, video surveillance, face recognition-based identification, and etc.

However, there is still a lack of flexible and scalable solution to detect the misbehaviour of the examinees in online examinations. Various methods have been proposed, such as face recognition, fingerprint verification etc. However, it is difficult to apply these methods to detect the misbehaviour of examinees in online examination. A normal examinee always looks at the computer screen and puts his/her hands on the desktop. While the examinee in abnormal state might look around or whisper, and even bows his/her head and checks up with prohibited documents.

Since the misbehaviour of examinees usually shows some changes in gesture, we propose a gesture-based solution to detect the misbehaviour which system webcam is adopted to detect the examinee's gesture, and a detection engine is developed to analyse the action events. In order to reduce false alarm ratio, a three-dimensional gesture detection scheme is proposed, both the duration and frequency of the detected action events are utilized to discriminate the target misbehaviour.

Finally, the detection engine reports the misbehaviour by matching the pre-defined misbehaviour patterns. Experiments demonstrate that our proposed solution can effectively distinguish the examinee’ misbehaviour from his/her normal action.

#### **System Requirements:**

To build Translator, your system must meet the following minimum requirements:

|  |  |
| --- | --- |
| **Processor:** | Minimum Intel i3 8TH Gen cocked at 2.5Ghz Processor: Intel® processor i5 8th gen at 2.5ghz or equivalent. |
| **Memory:** | RAM: 8 GB and Above. |
| **Disk Space:** | HDD: 8GB Minimum usage. for Model Training  SSD: Additional process and better performance. |
| **Operating System**: | Windows 8 or 10 (32 or 64-bit) (future Windows 11 is also compatible) |
| **GPU:** | Minimum Nvidia GTX 980M, recommended Nvidia GTX 1060 (VRAM-6GB) or equivalent |

|  |  |
| --- | --- |
| **Monitor**: | Monitor: 1980\*1080-pixel color monitor. |

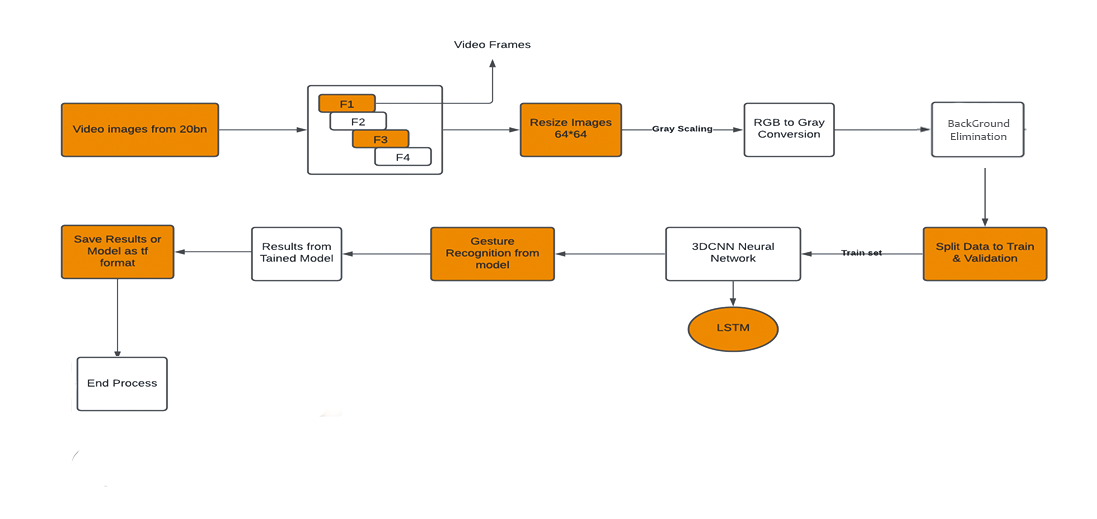
**Pre-processing:**

* Using self-generate images.
* As data are self-generated No use of Data cleaning.
* Resize the each Frame of Video Images to size of 64\*64 .
* Converting RGB images to Gray to and Background elimination method to process images to Model.
* Next step is to Split data into Train and Valid using SK-learn.
* Next step is to build a 3dcnn and LSTM Model for Gesture recognition.

**Proposed Model:**

For the gesture recognition process, it is challenging to learn both spatial and temporal features with feature extraction method. To address this challenge, we proposed an 3DCNN model with LSTM as seen in Figure below. The proposed architecture consists of several processes such as data collection, data pre-processing, training, and testing model to achieve our purpose. Under this section, we explain all these processes of our proposed system in detail.

**Model Training Process Diagram :**



**About CNN:**

A convolution is an integration function that expresses the amount of overlap of one function as it is shifted over another function. Intuitively, A convolution acts as a blender that mixes one function with another to give reduced data space while preserving the information.

**In terms of Neural Networks and Deep Learning:**

Convolutions are filter (matrix / vectors) with learnable parameters that are used to extract low-dimensional features from an input data.

They have the property to preserve the spatial or positional relationships between input data points

Convolutional neural networks exploit the spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers.

Intuitively, convolution is the step of applying the concept of sliding window (a filter with learnable weights) over the input and producing a weighted sum (of weights and input) as the output. The weighted sum is the feature space which is used as the input for the next layers.

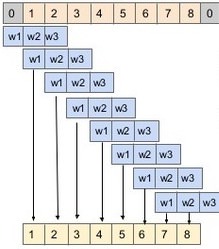
For example, in Face Recognition problem, first few convolution layers learn the presence of key points in the input image, next convolution layers learns the edges and shapes, and final convolution layers learns the face. In this example, the input space is first reduced to lower dimensional space (representing information about points / pixels), then this space is reduced to another space containing (edges / shapes) and finally it is reduced to classify faces in the images. Convolutions can be applied in N dimensions.

### **Types of Convolutions:**

Let's discuss what are different types of convolutions

### **1D Convolutions:**

Most simplistic convolutions are 1D convolutional are generally used on sequence datasets (but can be used for other use-cases as well). They can be used for extracting local 1D sub sequences from the input sequences and identify local patterns within the window of convolution. The following image shows how a 1 D convolution filter is applied to a sequence to obtain new features. Other common uses of 1D convolutions are seen in the area of NLP where every sentence is represented as a sequence of words.

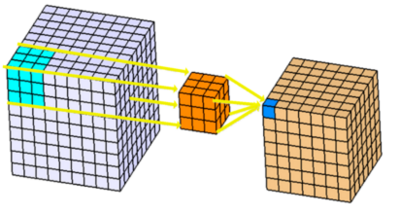


### **2D Convolutions:**

On image datasets, mostly 2D Convolutional filters are used in CNN architectures. The main idea of 2D convolutions is that the convolutional filter moves in 2-directions (x,y) to calculate low dimensional features from the image data. The output shape is also a 2-dimensional matrix.

**3D Convolutions:**

3D convolutions apply a 3-dimensional filter to the dataset and the filter moves 3-direction (x, y, z) to calculate the low-level feature representations. Their output shape is a 3-dimensional volume space such as cube or cuboid. Convolution where the kernel slides in 3 dimensions as opposed to 2 dimensions with 2D convolutions. One example use case is medical imaging where a model is constructed using 3D image slices. Additionally, video-based data has an additional temporal dimension over images making it suitable for this module. They are helpful in event detection in videos, 3D medical images etc. They are not limited to 3d space but can also be applied to 2d space inputs such as images.



**Use of 3D Convolutions Neural Network:**

Gesture recognition in computer science and language translation is the means of recognizing gestures through mathematical methods. Gesture recognition has become one of growing fields of research. gesture recognition has ample number of applications including human–computer interaction, sign language and virtual/augmented gaming technology. Users can perform gestures to control or interact with devices without physically touching them. There are many architectures designed in the field of gesture detection, but existing traditional solutions are not robust to detect gestures with high accuracy in real time in the presence of complex patterns in performing gestures. In this paper, we present a fast and efficient algorithm for classifying different face pose gestures using 3D-convolution neural networks. Unlike 2D-convolution neural networks, 3D-convolution networks extract features along the temporal dimension for analysis of gestures performed in videos. The paper also focuses on improving accuracy and describes data pre-processing and optimization techniques for obtaining the model inference in real time at 30fps. Our method achieves a correct recognition accuracy of 90.7% for the evaluation made on the testing videos in Realtime using Webcam Continuous. The detection process can be tested on laptops with standard specifications.

**LSTM Model:**

LSTM network models are a type of recurrent neural network that are able to learn and remember over long sequences of input data. They are intended for use with data that is comprised of long sequences of data, up to 200 to 400 time steps. They may be a good fit for this problem.

The model can support multiple parallel sequences of input data, such as each axis of the accelerometer and gyroscope data. The model learns to extract features from sequences of observations and how to map the internal features to different activity types.

The benefit of using LSTMs for sequence classification is that they can learn from the raw time series data directly, and in turn do not require domain expertise to manually engineer input features. The model can learn an internal representation of the time series data and ideally achieve comparable performance to models fit on a version of the dataset with engineered features.

**3DCNN with LSTM:**

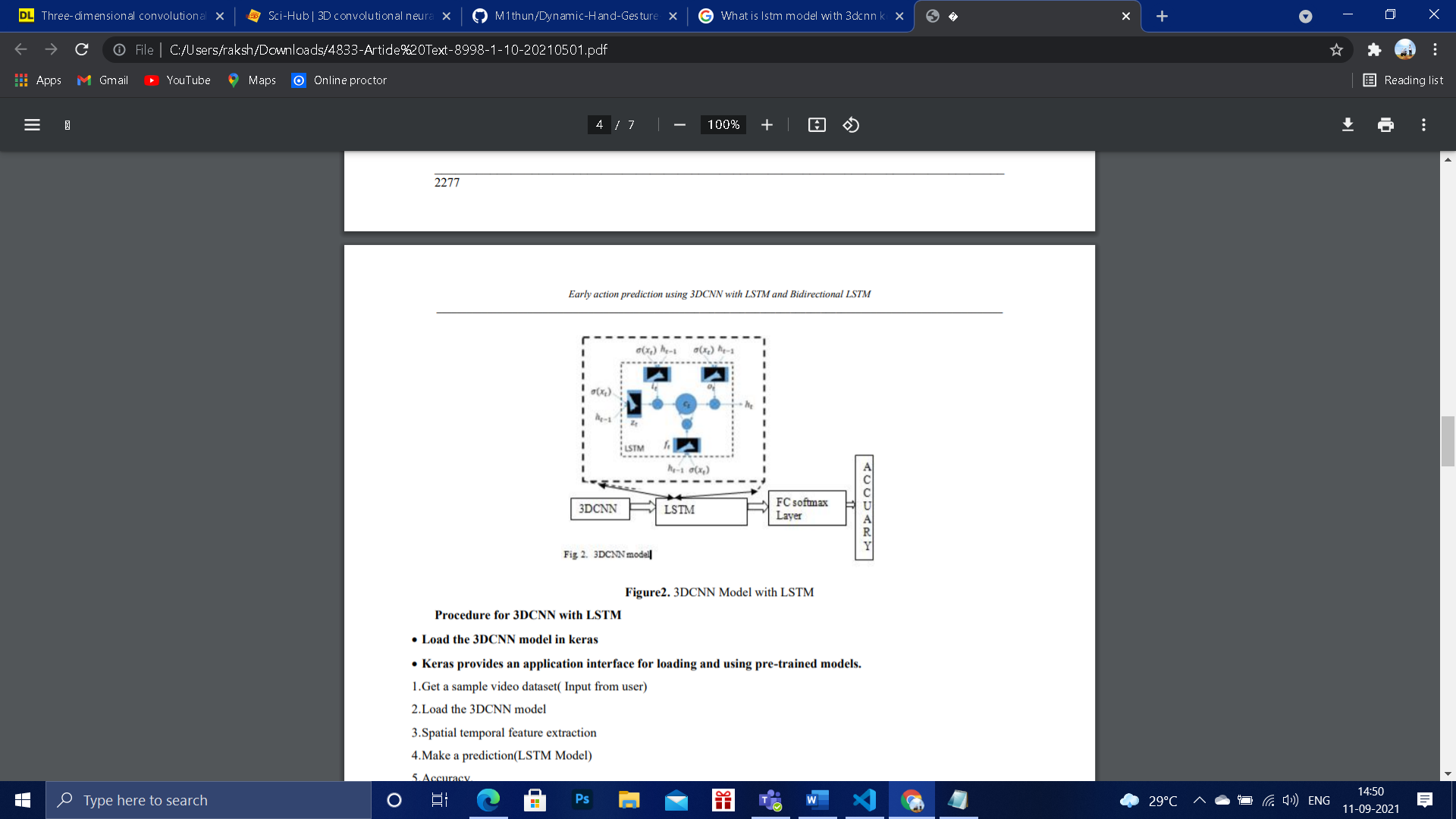
Compared to an image, and video is a stack of frames, for one second of video we get 30 frames per second so for a dataset pre-processing on video data is done by cropping them to a fixed size length which is problematic because in sequence learning, need to deal with variable length sequences. Therefore, to classify a 30 sec video with a 45 sec video. The 3DCNN, take advantage of CNN’s convolutional neural networks across different time scales. In this process from each frame CNN helps to extract frame features in the video.

The output of the action classification is given as an input to the LSTM model. There are 20 layers in this network. Convolutional layers are 12, followed by 5 pooling layers and one layer for FC, LSTM and output. The convolution block with paired with two or three 2D CNN and a pooling layer. It is followed by a dropout layer. The dropout layer has a dropout rate of 25%. Features are extracted using convolutional layer with 3\*3 kernel.

RELU activation function is used in the convolutional layer. The input dimension of image is reduced using max pooling layer with 2 × 2 kernels. LSTM is at last to extract time information. The output shape after convolution is (none, 7, 7, 512). The input size of LSTM layer becomes (49,512) due to reshape method. After analysing the time characteristics, the architecture sorts the video frames through a fully connected layer to predict whether they belong under any of the two categories (Abnormal/Normal). LSTM is an adapted version of recurrent neural networks to solve the problem of vanishing gradient. LSTM has a memory unit.

This memory unit encodes the knowledge learnt. It learns when to forget and update hidden states when new information is provided as input. Memory unit functionality is controlled by three gates: input gate, forget gate and output gate. The update and output functions are defined as below.

Input mapping sigmoid nonlinearity function. is the matrix representing the parameters of the gates? represents product operation with values of gate. LSTM control multiple gates to mitigate vanishing gradient problem and capture temporal dependencies. The 3DCNN model with LSTM is given in below Figure.



3DCNN Model with LSTM Procedure for 3DCNN with LSTM • Load the 3DCNN model in Kera’s

Kera’s provides an application interface for loading and using pre-trained models.

1.Get a sample video dataset (Input from user)

2. Load the 3DCNN model

3.Spatial temporal feature extraction

4.Make a prediction (LSTM Model)

5.Accuracy.

**Model Diagram:**

Diagram

Description automatically generated

**Model Summary:**

Model: "conv3d\_model"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv1 (Conv3D) multiple 896

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d (MaxPooling3D) multiple 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1 (Conv3D) multiple 55360

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_1 (MaxPooling3 multiple 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv\_lst\_m2d (ConvLSTM2D) multiple 149920

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten (Flatten) multiple 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

d1 (Dense) multiple 737408

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

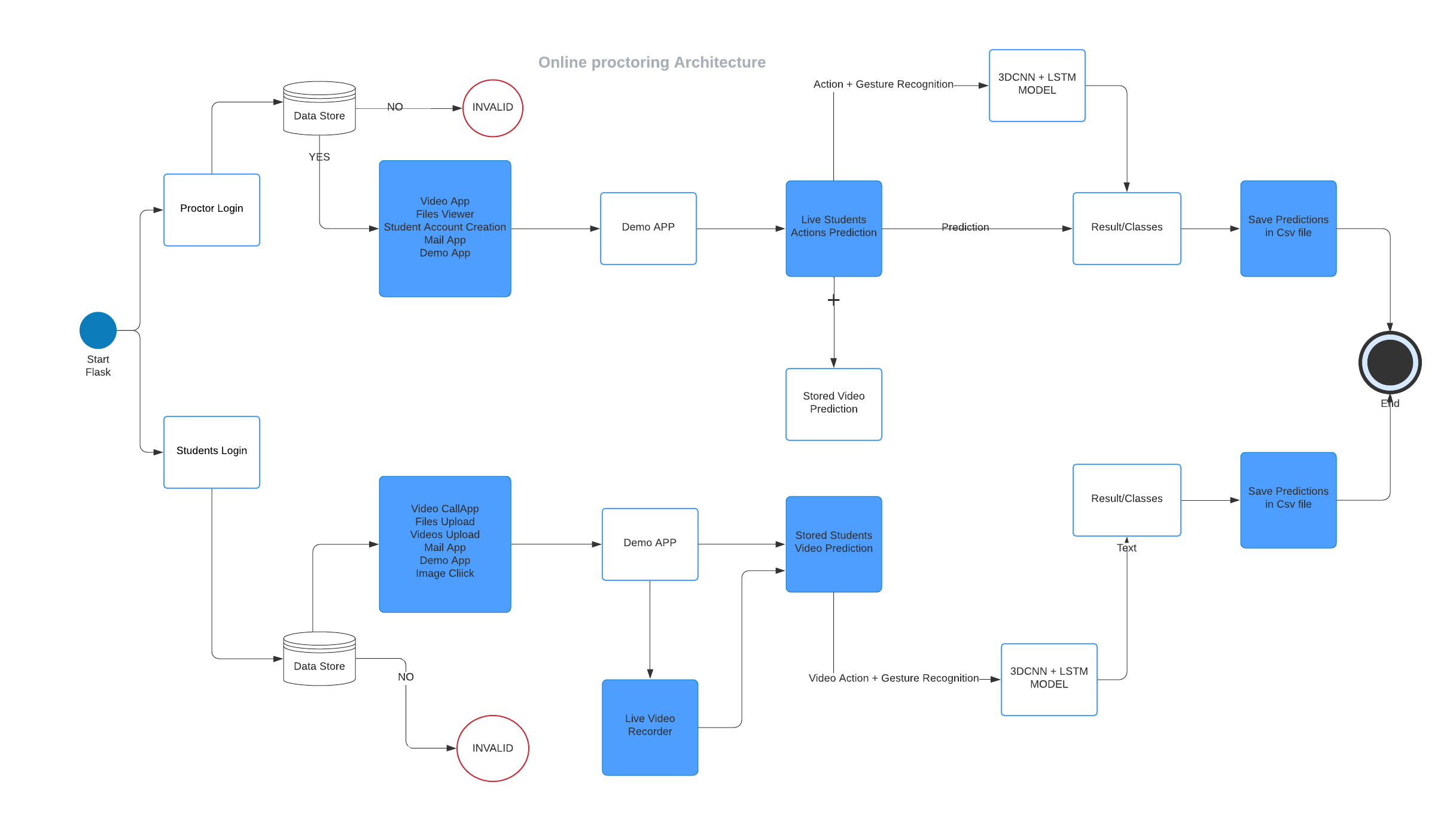
output (Dense) multiple 387

=================================================================

Total params: 943,971

Trainable params: 943,971

Non-trainable params: 0

**Project Workflow:**

Remote proctoring is the process of authenticating, authorising, and controlling an online test in a scalable manner. An invigilator must be present at the exam centre to certify participants taking the exam in the conventional exam method. Now that we know exactly what online examinations imply, let's look at how they operate.

**Examine:**

The instructor establishes an account and uploads a test.

The test taker establishes a profile and uploads information.

The test taker has completed their system's prefight check.

**Authenticate & Secure:**

ID based Login System for Examine

Google Oath Authentication for proctor

PROCTOR IN REAL TIME Exam is monitored by the vendor to ensure academic integrity.

RECORDING AND REVIEW Vendor proctor reviews the recording and identifies any integrity problems.

**Proctor:**

The proctor verifies the exam taker's ID (e.g., photo ID compared to test taker).

The proctor certifies that there are no prohibited items or aids in the testing area (webcam view of the room, malpractice gesture predictions).

**Report**:

Incidents are reported back to the authorities by send mail or warning (flagged by priority).

Test taker feedback is provided

This is a summary of the normal workflow for both Proctor and Examiner.

**References:**

[**https://github.com/PrettyPrinted/building\_user\_login\_system**](https://github.com/PrettyPrinted/building_user_login_system)**.**

[**https://github.com/anasmorahhib/3D-CNN-Gesture-recognition**](https://github.com/anasmorahhib/3D-CNN-Gesture-recognition)**.**

[**https://github.com/waynshang/Gesture-Recognition-with-3DCNN**](https://github.com/waynshang/Gesture-Recognition-with-3DCNN)**.**

**Gesture based Misbehaviour Detection in Online Examination : DOI :** [**https://sci-hub.se/10.1109/ICCSE.2016.7581586**](https://sci-hub.se/10.1109/ICCSE.2016.7581586)**.**

**Early Action Prediction using 3DCNN with LSTM and Bidirectional LSTM**

[**https://towardsdatascience.com/camera-app-with-flask-and-opencv-bd147f6c0eec**](https://towardsdatascience.com/camera-app-with-flask-and-opencv-bd147f6c0eec)**.**

[**https://blog.miguelgrinberg.com/post/video-streaming-with-flask**](https://blog.miguelgrinberg.com/post/video-streaming-with-flask)**.**

[**https://pythonhosted.org/Flask-Mail/**](https://pythonhosted.org/Flask-Mail/)**.**

[**https://realpython.com/flask-google-login/**](https://realpython.com/flask-google-login/)