MULTI-LABEL BODY LANGUAGE RECOGNITION

Using Computer Vision and Deep Learning Models FOR EXAM MONITORING

Sarit Mukhopadhyay|Govinda K| SCOPE

Introduction

In the modern era, the rapid development of AI technology has introduced various classes of IOT devices such as smart watches, wireless earphones and smart glasses just to name a few. Even more are being used in the industrial sector, where a control center monitors and regulates different components such as sensors, regulators, breakers, switches to maintain an efficient assembly line so that productivity is maximized. This makes activity monitoring and data collection crucial to the advancement of connectivity within a control system. One of the most basic control systems in almost any part of the world are educational institutions, where each student can be monitored to optimize their individual potential and promote growth and acceleration of future human advancement.

Motivation

Computer Vision software's on mobile applications from Facebook, Instagram, Tik-Tok or Snapchat have been exploding across the social network platforms. Recognizing different actions from a given sequence of visual observations from initial and final state conditions provide basis of parameters for neural networks to learn and train sequential models However, reliable and precise vision-based systems still remain to be one of the biggest challenges in the field of computer visions. To further the advancement of this field, a vast proportion of research needs to be fast-tracked through multi-modality-based recognition systems that are fluid and cohesive within each application that requires interactions between human and computer vision systems.

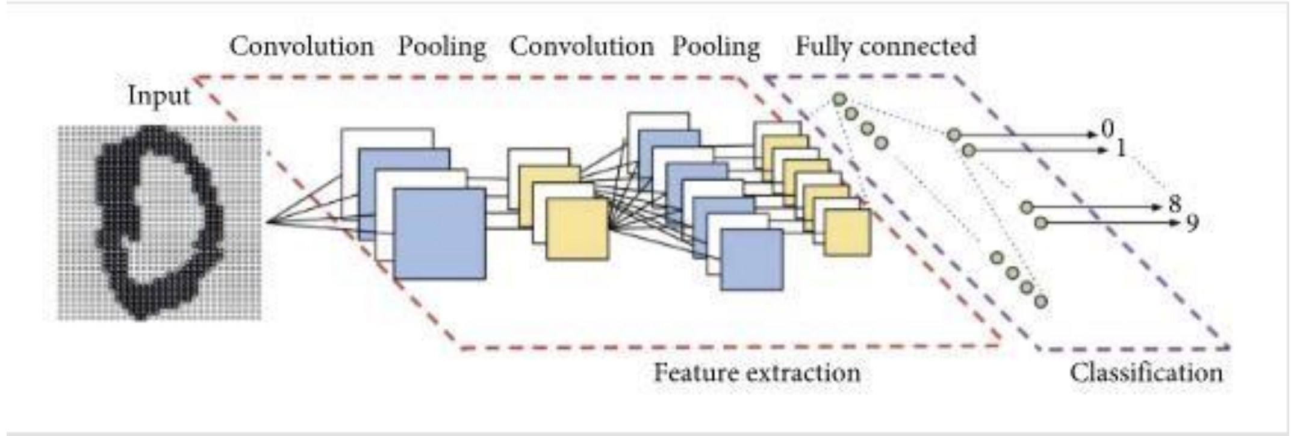
SCOPE of the Project

Using computer vision and deep learning models to standardize online test monitoring and make it affordable for almost anyone to use. The Two main Objectives are:

1.Use computer vision detection systems classify certain actions in a testing Scenario such as covering of the face or raising the hand for questions. This also includes data collection and preprocessing and data augmentation.

2.To implement deep neural networks to actively and accurately classify the collected data to the respective actions, exploring CNN, RNN, LSTM and Dense neural nets and Finally Test in real time video detection from webcam

Methodology



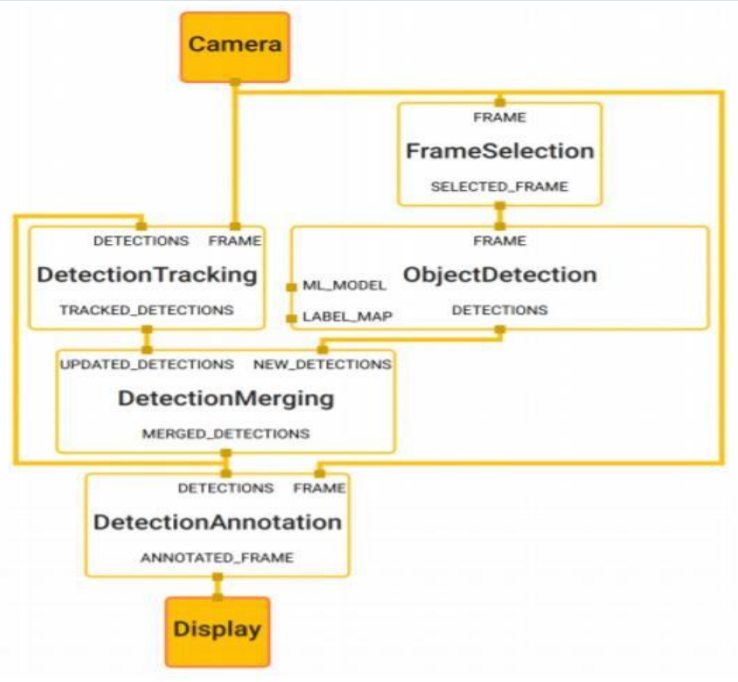


figure 1: shown above is the proposed methodology for the feature extraction and classification. Figure 2 on the right shows the proposed capturing sequence.

The collection involved simulating the real time data points to emulate the actual behaviour to distinguished action classes. These points are then saved using the OS library to make 3 folders for 3 classes and storing 90 sequences with each sequence containing 30 frames each. When collecting the data, it was detrimental to the accuracy of the model that the sequence of the frames was cohesive with the actions that were intended to be emulated in the real world. However, it was also hard to perfectly emulate the real-world scenario as there were limitations to the CV data point recognition system. As there was a lot of latency in the actual detection and accuracy of the face, pose and hand estimation models.

This made the collection of the data points quite difficult, as it was key to make sure that the data points were visible enough so that it would detect and record the positions in three dimensions. One the training data sequences are modelled using the LSTM and Dense Neural Networks, they were tested using variations of training and testing set parameters to find the best producing recognition model. This h5 model was then tested in real time detections to be evaluated for its recognition accuracy, processing rate and time. Numerous hyper parameters were to be estimated in real time to study which features were most pertinent towards attaining a reliable detection model

```
<matplotlib.image.AxesImage at 0x2745dee18e0>
```

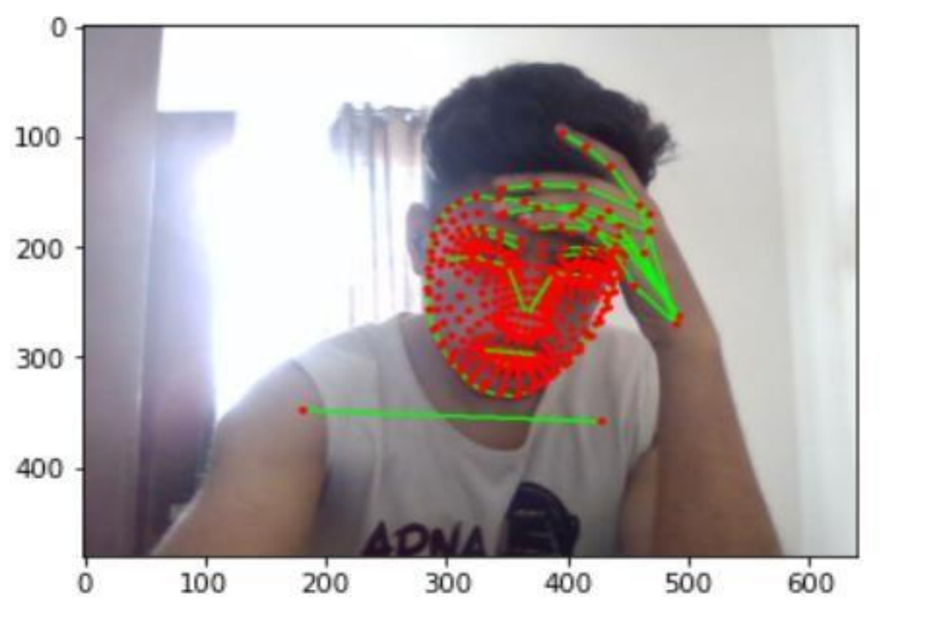
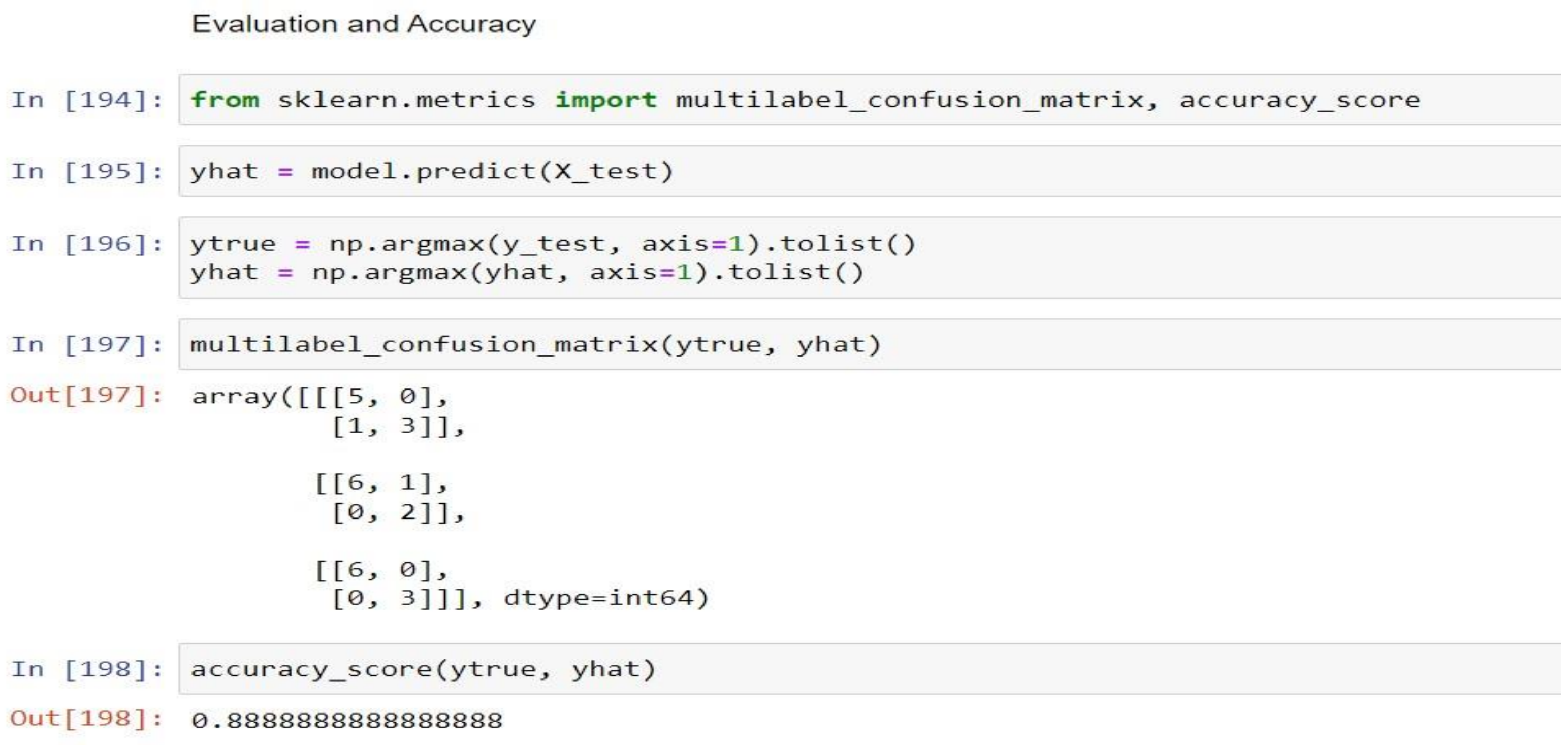


Figure 3 -Tools used:
1. Anaconda Distribution system
2. Jupyter Notebook to document the program, and analyzing process
3.Spyder for the Data collection process.
4.And the following Python Frameworks – TensorFlow, Keras, OpenCV and MediaPipe

Results

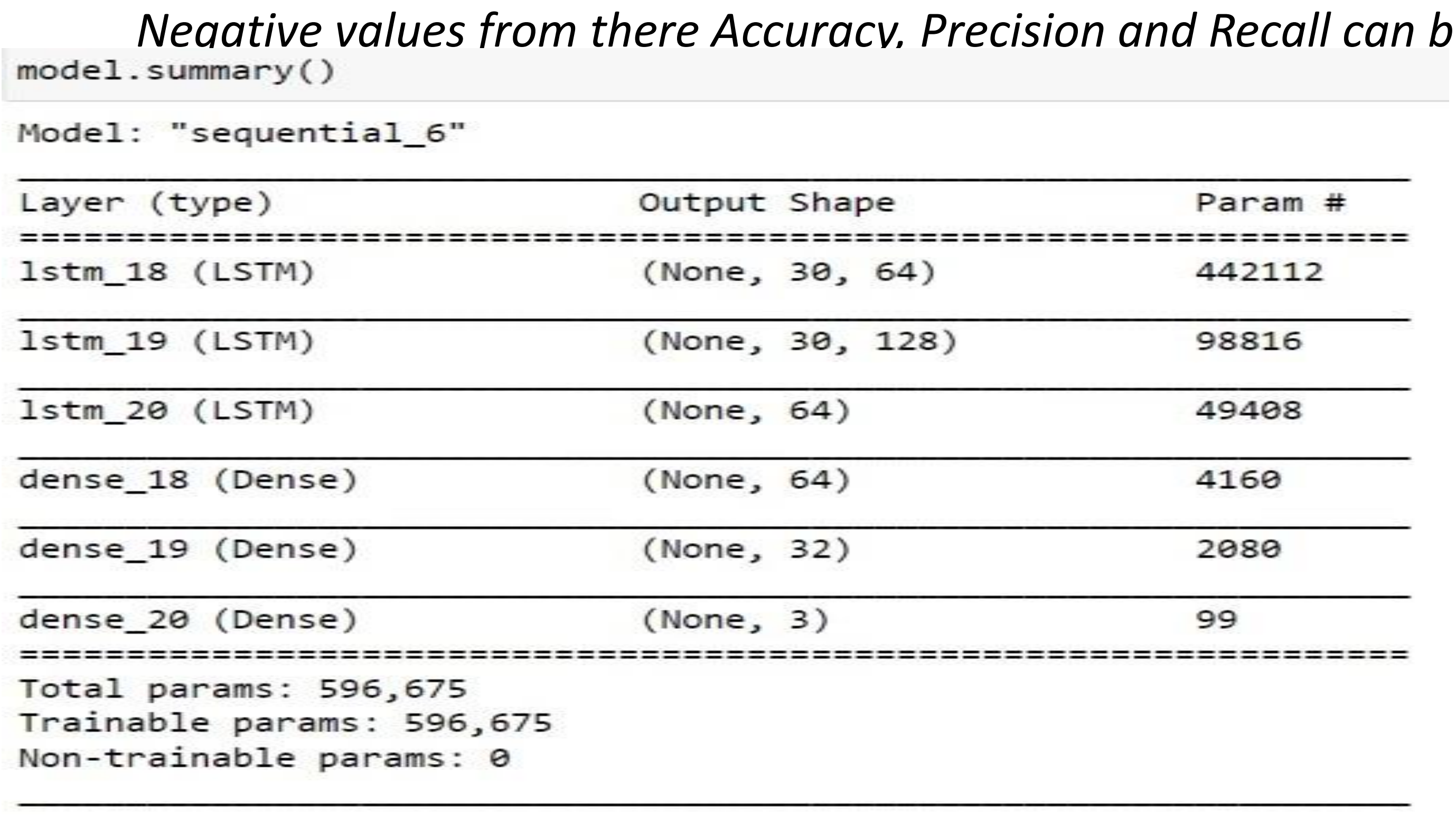
Once the training data sequences are modelled using the LSTM and Dense Neural Networks, they were tested using variations of training and testing set parameters to find the best producing recognition model. This h5 model was then tested in real time detections to be evaluated for its recognition accuracy, processing rate and time. Numerous hyper parameters were to be estimated in real time to study which features were most pertinent towards attaining a reliable detection model. From the deep learning model we built from the limited data sets extracted, we achieved an accuracy of approximately 89% . The resulting accuracy is not bad but with more time and cleaner data sets the accuracy should also increase.



Evaluation and Accuracy

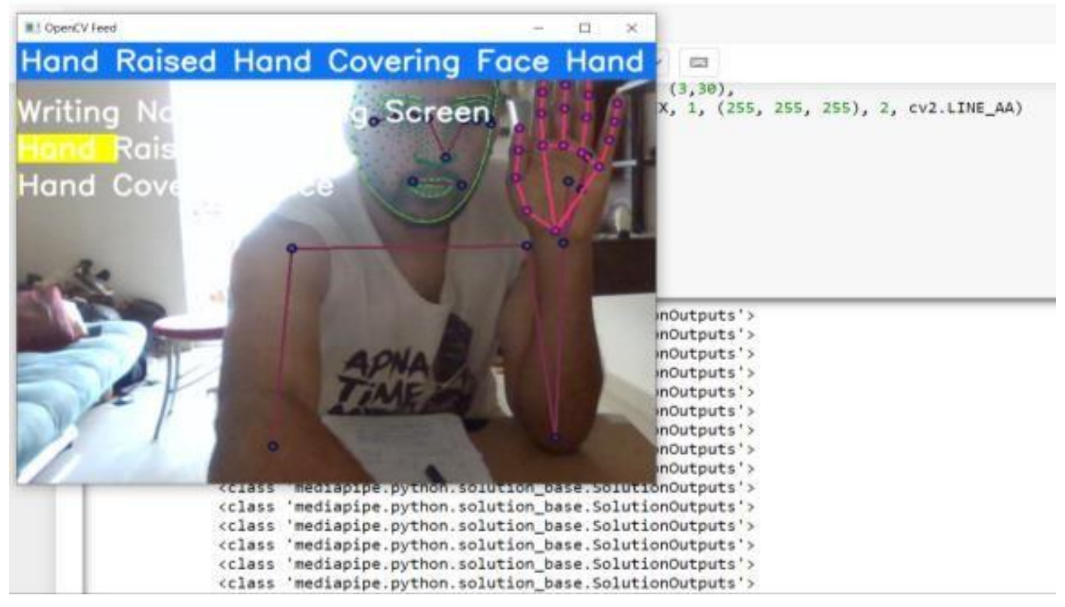
```
In [194]: from sklearn.metrics import multilabel_confusion_matrix, accuracy_score

In [195]: yhat = model.predict(X_test)

In [196]: ytrue = np.argmax(y_test, axis=1).tolist()
          yhat = np.argmax(yhat, axis=1).tolist()

In [197]: multilabel_confusion_matrix(ytrue, yhat)
Out[197]: array([[[5, 0],
                  [1, 3]],

                 [[6, 1],
                  [0, 2]],

                 [[6, 0],
                  [0, 3]]], dtype=int64)

In [198]: accuracy_score(ytrue, yhat)
Out[198]: 0.8888888888888888
```

Figure 4: Above Shows the Confusion matrix containing the True/False positive and Negative values from there Accuracy, Precision and Recall can be found



```
model.summary()

Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_18 (LSTM)               (None, 30, 64)            442112
_________________________________________________________________
lstm_19 (LSTM)               (None, 30, 128)           98816
_________________________________________________________________
lstm_20 (LSTM)               (None, 64)                49408
_________________________________________________________________
dense_18 (Dense)             (None, 64)                4160
_________________________________________________________________
dense_19 (Dense)             (None, 32)                2080
_________________________________________________________________
dense_20 (Dense)             (None, 3)                 99
=================================================================
Total params: 596,675
Trainable params: 596,675
Non-trainable params: 0
_________________________________________________________________
```

Figure 5: Above shows the output shape and parameters of the 3 LSTM Layers and 3 Dense Layers



```
(3,30),
X, 1, (255, 255, 255), 2, cv2.LINE_AA)
nOutputs'>
nOutputs'>
nOutputs'>
nOutputs'>
nOutputs'>
nOutputs'>
nOutputs'>
nOutputs'>
nOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
```

Figure 5: shows the actual results of the program working in real time, Identifying as the user raises his hands

Conclusion

The framework discussed in this paper helps in leading assessments by fair means and henceforth, keeps up with its uprightness. This review shows how to try not to cheat in web-based assessments by utilizing semi-robotized administering in view of vision what's more, sound capacities, as well as observing a few understudies without a moment's delay. In any case, on the off chance that there is an individual sitting behind the PC, the understudy can speak with that individual by perusing the inquiry. This can be taken special care of by having a 360-degree camera checking the entire room of the understudy. For a future online test setting, a person could easily retrieve information from either their smart watch or glasses even if their phone and laptops were being monitored. This makes early data collection crucial to understanding testing environments and different ways people are taking tests.

References

1. Aitik Gupta ABV-Indian Institute of Information Technology and Management, et al. "ActiveNet: A Computer-Vision Based Approach to Determine Lethargy: 8th ACM Ikdd Cods and 26th COMAD." *ACM Other Conferences*, 1 Jan. 2021, https://dl.acm.org/doi/10.1145/3430984.3430986.

2. Zhang, Fan, et al. "MediaPipe Hands: On-Device Real-Time Hand Tracking." *ArXiv.org*, 18 June 2020, https://arxiv.org/abs/2006.10214v1.

3. A. S. Nikam and A. G. Ambekar, "Sign language recognition using image based hand gesture recognition techniques," 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 2016, pp. 1-5, doi: 10.1109/GET.2016.7916786. https://ieeexplore.ieee.org/document/7916786

4. Lugaresi, Camillo, et al. "MediaPipe: A Framework for Building Perception Pipelines." *ArXiv.org*, 14 June 2019, https://arxiv.org/abs/1906.08172.