**Report**

**Final Project**

ON

**“Human action recognition model”**

**Submitted by**

Kanishk Mehta – Data Analysis Lead

PrahaladhChandrahasan – Modelling Lead

Sairaj Bhise – Deployment Lead

**Guided by**

Yasin Shah

**Technocolabs**

(Build with AI)

**Indore, Madhyapradesh**

**Modules**

|  |  |  |
| --- | --- | --- |
| **Module** | **Name** | **Lead** |
| 1 | Analysis Stage | Kanishk Mehta |
| 2 | Modelling and Evaluation | Prahaladh Chandrahasan |
| 3 | Deployment | Sairaj Bhise |

**Project Deployment**

Deploying a **Deep learning** model, known as model deployment, simply means to integrate a **Deep learning** model and integrate it into an existing production environment where it can take in an input and return an output.

1) **Framework - Flask**

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

2) **Model type – HDF5**

**The HDF5 technology suite includes:**

* A versatile data model that can represent very complex data objects and a wide variety of metadata.
* A completely portable file format with no limit on the number or size of data objects in the collection.
* A software library that runs on a range of computational platforms, from laptops to massively parallel systems, and implements a high-level API with C, C++, Fortran 90, and Java interfaces.
* A rich set of integrated performance features that allow for access time and storage space optimizations.
* Tools and applications for managing, manipulating, viewing, and analyzing the data in the collection.

3) **Sub Modules - (In total 4)**

* Taking input from user.
* Extracting video and tensors.
* Feeding tensors to model and making prediction.
* Showing prediction to user.

**Detailed Steps**

**Step 1: Taking input**

* Input is in the form of link of YouTube video (Max 10 sec).
* Input will be taken from html form connected to flask as backend.
* The link will be stored as string in variable.

**Step 2: Extracting Video and Tensors**

* The link will be the sent to module named make\_tensors.py.
* Where it will be passed through the script where the video gets extracted and saved in current directory.
* Then the path of the video will be given to the module named utils.py, where the read\_video method will generate the tensors and store it in variable.

**Step 3: Making Prediction**

* After we got the tensors, we import our model (final\_model.h5).
* We feed the tensors to model and make predictions and store it in variable.
* Then we extract the activity name using data frame of targets that we imported earlier and pass it to main app.

**Step 4: Showing prediction to user**

* After we got the predictions and activity name, we give argument to the render template method where replace the answer holding variable.