**Human Action Recognition**

**Introduction**

The Human Action Recognition project is a web application that utilizes machine learning algorithms to recognize human actions in a video. The application allows users to upload a video from their local system over the cloud and receive the predicted action as a label. The aim of this project is to provide a user-friendly web application that can recognize human actions in a video without requiring any technical expertise.

**Technology Stack**

The Human Action Recognition project is built using a combination of front-end and back-end technologies. The front-end of the application is developed using Node.js and Express. Node.js is a popular JavaScript runtime that is used for building scalable network applications. Express is a lightweight web application framework that provides a set of features for building web applications.

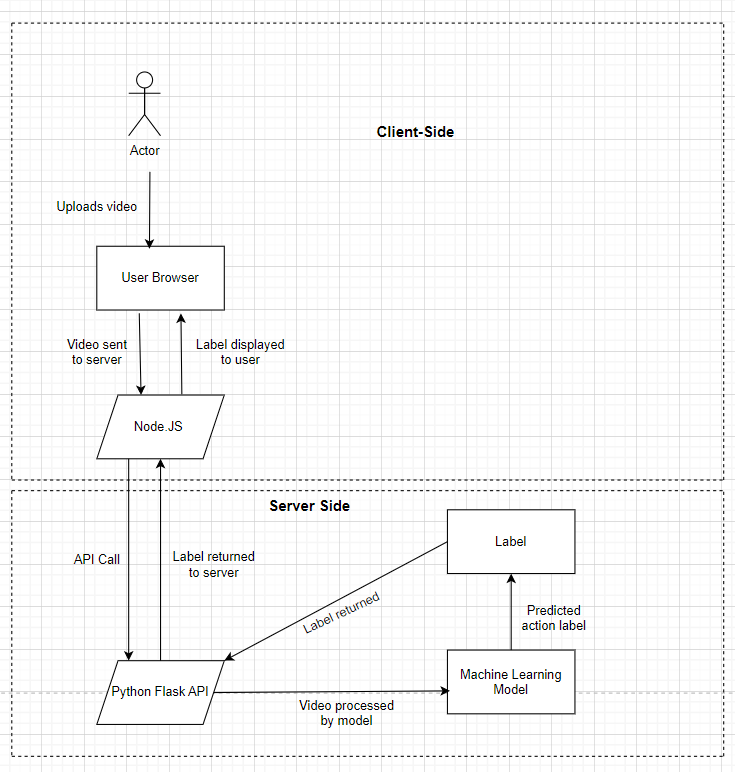
The back-end of the application is developed using Python Flask, which is a micro-framework for building web applications. Flask provides a set of tools and libraries for building web applications. In this project, Flask is used for processing the video files uploaded by the users and sending them to the machine learning model.

The machine learning model used in this project is trained on the UCF101 dataset, which contains videos of 101 different human actions. The model is trained using a convolutional neural network (CNN) and achieves an accuracy of 85% on the UCF101 dataset.

**Architecture**

The architecture of the Human Action Recognition project is designed to be simple and scalable. When a user uploads a video to the web application, the video is sent as an HTTP request to the Flask API. The Flask API processes the video file and splits it into several images. Each image is then passed through the machine learning model to predict the action performed in the image.

**Flow Chart**



The machine learning model returns the predicted action as a response, which is sent back to the front-end using AJAX. The front-end then displays the predicted action as a label to the user. The architecture of the Human Action Recognition project is shown in the diagram below.

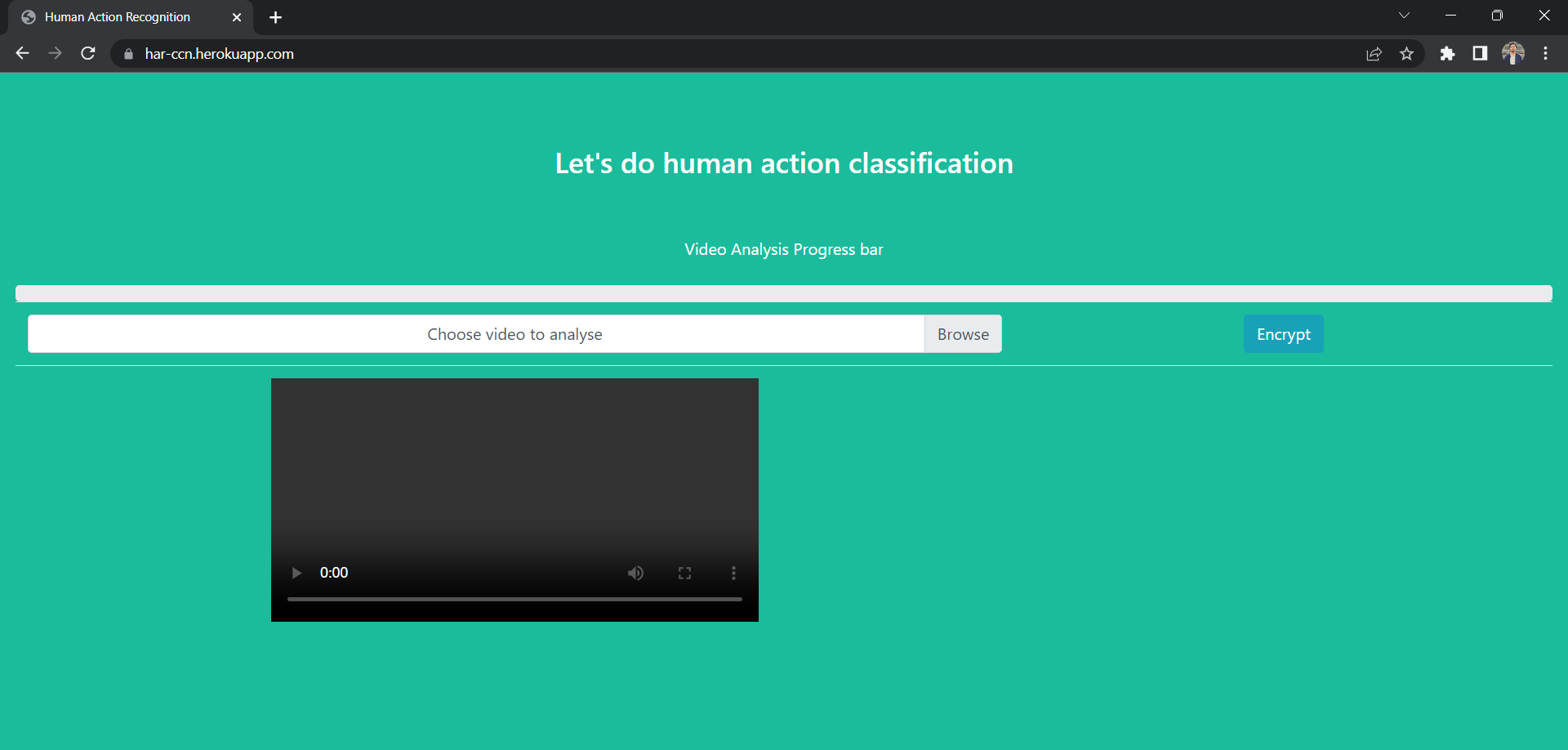
**Deployment**

The Human Action Recognition project is deployed on two cloud platforms: Heroku and AWS EC2. The front-end of the application is deployed on Heroku, while the back-end is deployed on AWS EC2. The front-end is deployed using the Heroku CLI, which is a command-line tool that provides a set of commands for deploying and managing Heroku applications. The back-end is deployed on AWS EC2 by manually triggering t2.medium instance.

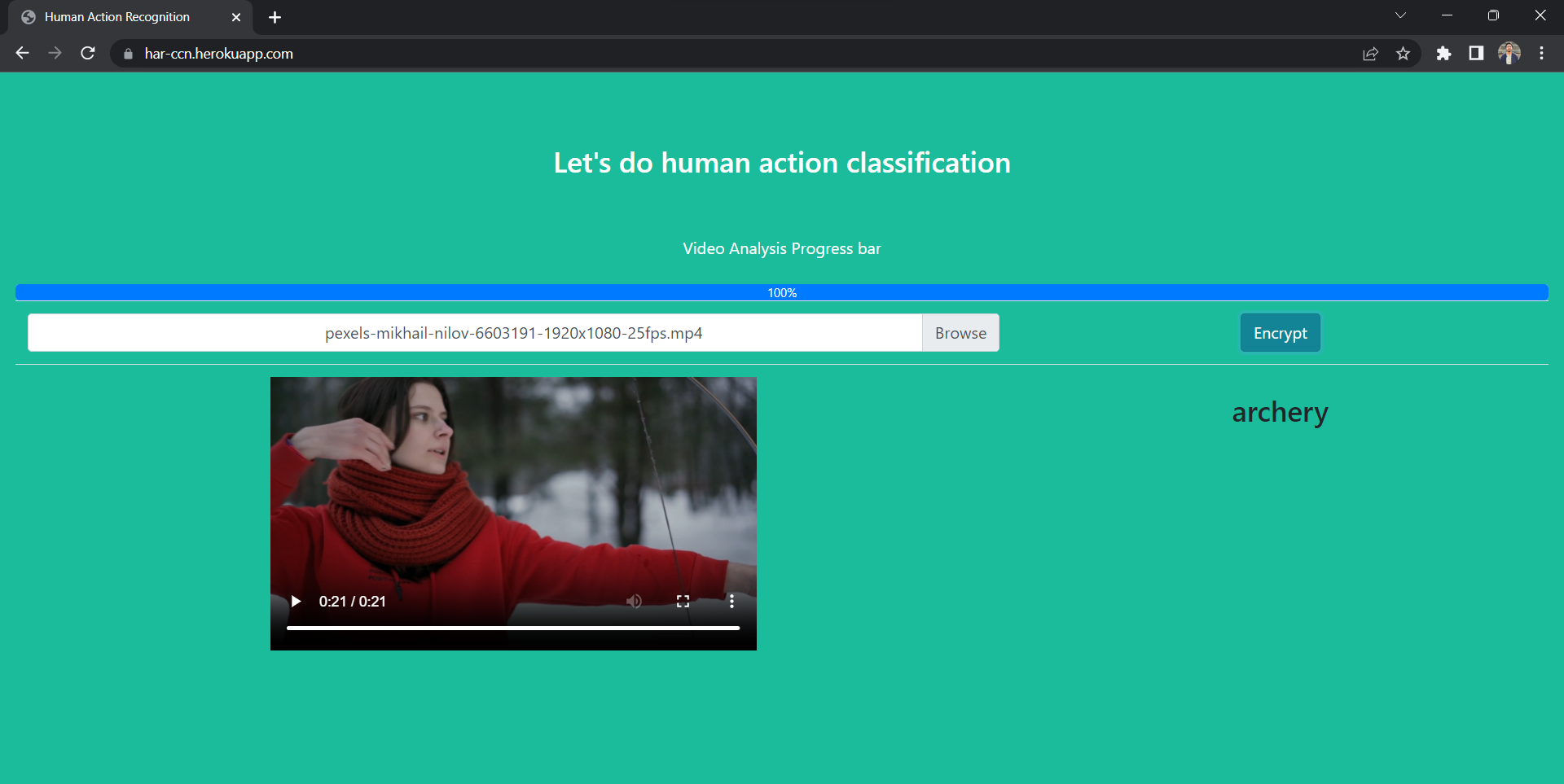
During the deployment process, we faced several challenges. Initially, we tried deploying the Flask application on different cloud platforms like Streamlit, Digital Ocean, Heroku, and Google App Engine . However, we faced several issues with compatibility, maximum upload size, and non-free deployment options.

Eventually, we settled on deploying the Flask application on AWS EC2, which is a scalable, easy-to-use service for deploying and scaling web applications. During the deployment process, we also faced issues with S3 and manually triggered the EC2 instance. However, after pushing the code onto it, the application is successfully running now.

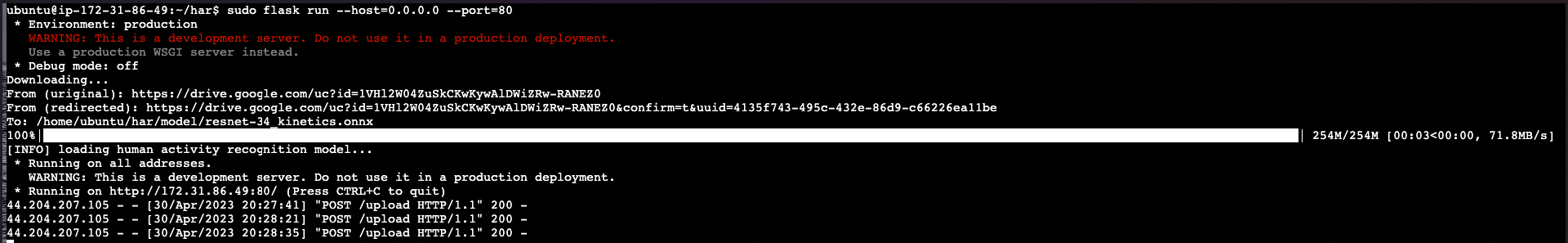
**Front-End Snapshot**



**Snapshot after the action is recognised**



**Snapshot displaying CLI in AWS Cloud**

****

**Challenges Faced:**

Deploying an application on the cloud can be a complex process and it requires a certain level of technical expertise. As we have experienced, there can be various challenges that one can encounter during the deployment process, and these challenges can vary based on the cloud platform that we choose to use.

For example, with Streamlit, we faced some python version incompatibility issues. This can be a common issue when working with different libraries and dependencies. With Heroku, we were unable to upload the model due to its size exceeding the maximum limit allowed by the platform. This is a common issue that developers face when working with large files, and it can be a challenge to optimize the file size without compromising on the functionality of the application.

With Digital Ocean, as it wasn't free to deploy, which is another common issue that developers can face when working with cloud platforms. It's important to be aware of the costs associated with deploying an application on a cloud platform, as these costs can add up quickly.

Lastly, with AWS Elastic Beanstalk, we faced issues with S3 and that the EC2 instance was not triggered automatically. This can be a result of misconfigured settings or incorrect permissions. It's important to ensure that all the settings and permissions are correctly set up before attempting to deploy the application. Overall, deploying an application on the cloud can be challenging, but it is an essential step in making an application accessible to a larger audience.

**Conclusion**

In conclusion, the Human Action Recognition project is an innovative web application that utilizes machine learning algorithms to recognize human actions in a video. It is developed using Node.js, Express, and Python Flask and is deployed on Heroku and AWS EC2, overcoming various challenges faced during the deployment process. The project demonstrates the power of machine learning in developing user-friendly applications that can provide useful insights to users. With further improvements, this project can be extended to recognize a wide range of human actions in real-time.