EAI 6010 Module 5 : Facial Landmark Detection Microservice: A Deployment Case Study

Course	EAI6010	**Module**	5
Submission By	Arjun Shrivatsan		
Included	Assignment Report Link to Deployed Service Source Code Link		

Overview

In this project, I have developed and deployed a microservice that performs facial landmark detection using MediaPipe, OpenCV, and Flask. The service processes a user-uploaded image and returns a detailed JSON response of facial landmarks and derived facial measurements like forehead width, eye gap, and chin width. Facial landmark detection has wide-ranging applications in healthcare, augmented reality, and safety systems. The idea behind building this service stems from a real-world requirement in helmet manufacturing, where accurate facial dimensions can improve helmet fit and thereby enhance safety.

Use Case

I have chosen as part of my work that I have done in sports safety. The primary use case is for personalized helmet fitting, safety equipment design, or biometric analysis. By retrieving accurate facial measurements from a 2D image, manufacturers or developers can build safer and more customized wearables without requiring expensive 3D scanning hardware.

Why this Use Case

By using a lightweight and web-deployable model like MediaPipe's FaceMesh, I can offer scalable and accurate detection without intensive server-side computation. The use case also presents an intersection of AI, healthcare, and manufacturing—domains where I have practical experience. This deployment serves as a starting point to build future pipelines that include photogrammetry, 3D modeling, and reinforcement learning-based fit optimization.
This microservice can be integrated into systems that require accurate facial geometry measurements, such as custom safety gear fitting, biometric systems, and facial analysis for behavioral studies. I chose this use case as I have previously worked with helmet

manufacturers and observed the gap in custom-fit solutions. Real-time and mobile-based facial detection solutions can reduce dependency on expensive scanning equipment.

Approach and Technology Stack

The core approach was to create a RESTful API using Flask that accepts an image, processes it using MediaPipe's FaceMesh solution, and returns landmark coordinates and derived facial dimensions in JSON format. MediaPipe is developed by Google Research and is well-documented and efficient for real-time applications (Lugaresi et al., 2019). The service is deployed on Render.com, ensuring public accessibility with minimal server maintenance.

Tech Stack

Component	Details
Programming Language	Python 3.10
Web Framework	Flask
Face Detection	MediaPipe (FaceMesh)
Image Processing	OpenCV
Image Transformation	PIL (Python Imaging Library)
Deployment Platform	Render.com – for ease of use and Docker support
Containerization	Docker
API Testing Tool	cURL

Dataset

The dataset used for testing and demonstration in this microservice is derived from the **CelebA (CelebFaces Attributes Dataset)**, a widely used large-scale face attributes dataset with more than 200,000 celebrity images, each annotated with 40 attribute labels and 5 landmark locations. Although our model (MediaPipe FaceMesh) does not use the attribute annotations, the diversity and facial variation in CelebA make it an excellent dataset for testing face landmark detection tasks. It covers a wide range of pose variations, facial expressions, occlusions, and lighting conditions, providing robustness in evaluation.

The dataset used for testing the service includes high-resolution face images stored locally. These are personal test datasets and not publicly available. For actual deployment, any real face image with frontal orientation can be used to test the API.

We use a few sample images from the CelebA dataset as test inputs to validate the endpoint's ability to detect facial landmarks and measure facial features effectively.

Implementation

Development began with local development using Jupyter Notebooks and MediaPipe's visualization tools to test the FaceMesh model. I then built the Flask microservice and tested it locally with different image sizes and angles to ensure robustness. Unit tests were written to validate JSON responses, and integration testing was done using `curl` commands. The application was then Dockerized for consistent deployment across environments.

Output

The output from the `/face-coordinates` endpoint is returned in **JSON** format and includes two major components:

1. **Landmarks** – A list of facial keypoints with their `id`, and `x`, `y` pixel positions. These are generated by MediaPipe's FaceMesh model, which detects 468 facial landmarks per face. Each coordinate pair corresponds to a unique anatomical facial feature.
2. **Measurements** – Three key facial metrics computed using landmark positions:
   - `forehead_width_px`
   - `eye_gap_px`
   - `chin_width_px`

These values are helpful for various applications, including custom safety helmet design, avatar modeling, and biometric validation. The output is lightweight and easy to parse, making it ideal for further processing or integration with frontend visualization tools.

Here is a brief example snippet of the output:

```
json
Copy code
{
  "landmarks": [{"id": 0, "x": 91, "y": 147}, ...],
  "measurements": {
    "forehead_width_px": 10,
    "eye_gap_px": 21,
    "chin_width_px": 8
  }
}
```

This structured response helps users, including the instructor, clearly verify that the model is functioning as intended and delivering valid facial geometry data.

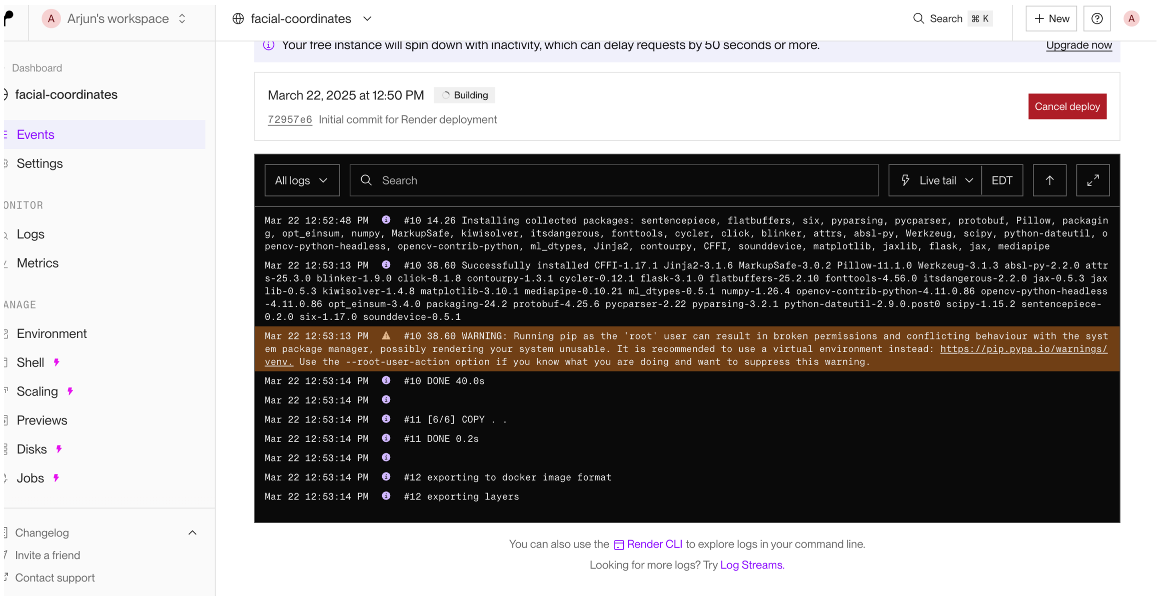
Implementation Details

Step / Component	Description
Endpoint	`/face-coordinates` (POST method)

Step / Component	Description
Input	Image uploaded as `multipart/form-data`
Image Handling	Converted to NumPy array and BGR format using OpenCV
Landmark Detection	MediaPipe FaceMesh model
Landmark Mapping	Facial feature indices mapped (e.g., eyes, forehead, chin)
Measurement Calculation	Pixel distance between key facial points
Output	JSON with:
	– `landmarks`: list of facial coordinates (`id`, `x`, `y`)
	– `measurements`: forehead width, eye gap, and chin width (in pixels)

Deployment

The deployment used Render.com with a connected GitHub repository. A Dockerfile was created to install dependencies from `requirements.txt` and expose the Flask server. After successful builds and logs verification, I tested the public endpoint using image uploads. The Render deployment remains live, and the endpoint can be invoked reliably as long as the Render service does not auto-suspend.



Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. Upgrade now

March 22, 2025 at 12:50 PM

72957e6 Initial commit for Render deployment

```
Mar 22 12:52:48 PM  #10 14.26 Installing collected packages: sentencepiece, flatbuffers, six, pyparsing, pycparser, protobuf, Pillow, packagin
g, opt_einsum, numpy, MarkupSafe, kiwisolver, itsdangerous, fonttools, cycler, click, blinker, attrs, absl-py, Werkzeug, scipy, python-dateutil, o
pencv-python-headless, opencv-contrib-python, ml_dtypes, Jinja2, contourpy, CFFI, sounddevice, matplotlib, jaxlib, flask, jax, mediapipe
Mar 22 12:53:13 PM  #10 38.60 Successfully installed CFFI-1.17.1 Jinja2-3.1.6 MarkupSafe-3.0.2 Pillow-11.1.0 Werkzeug-3.1.3 absl-py-2.2.0 attr
s-25.3.0 blinker-1.9.0 click-8.1.8 contourpy-1.3.1 cycler-0.12.1 flask-3.1.0 flatbuffers-25.2.10 fonttools-4.56.0 itsdangerous-2.2.0 jax-0.5.3 jax
lib-0.5.3 kiwisolver-1.4.8 matplotlib-3.10.1 mediapipe-0.10.21 ml_dtypes-0.5.1 numpy-1.26.4 opencv-contrib-python-4.11.0.86 opencv-python-headless
-4.11.0.86 opt_einsum-3.4.0 packaging-24.2 protobuf-4.25.6 pycparser-2.22 pyparsing-3.2.1 python-dateutil-2.9.0.post0 scipy-1.15.2 sentencepiece-
0.2.0 six-1.17.0 sounddevice-0.5.1
Mar 22 12:53:13 PM  #10 38.60 WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the syst
em package manager, possibly rendering your system unusable. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/
venv. Use the --root-user-action option if you know what you are doing and want to suppress this warning.
Mar 22 12:53:14 PM  #10 DONE 40.0s
Mar 22 12:53:14 PM
Mar 22 12:53:14 PM  #11 [6/6] COPY . .
Mar 22 12:53:14 PM  #11 DONE 0.2s
Mar 22 12:53:14 PM
Mar 22 12:53:14 PM  #12 exporting to docker image format
Mar 22 12:53:14 PM  #12 exporting layers
```

You can also use the Render CLI to explore logs in your command line.
Looking for more logs? Try Log Streams.

Annexure B: Key Information

Item	Details
Service URL	https://facial-coordinates.onrender.com
Git URL	[Private/Local]
Dataset Reference	Custom local test images
MediaPipe Reference	Lugaresi et al., 2019 - https://arxiv.org/abs/1906.08172
Tech Stack Required	Python 3.10, Flask, OpenCV, MediaPipe, Docker
Credentials Required	None for API access

Annexure C: Installation Instructions

1. Clone the repository.
2. Install dependencies: `pip install -r requirements.txt`
3. Run using Flask: `python app.py`
4. To deploy, use Docker and push to Render.com.
5. Test API with `curl` or Postman.

Additional Notes

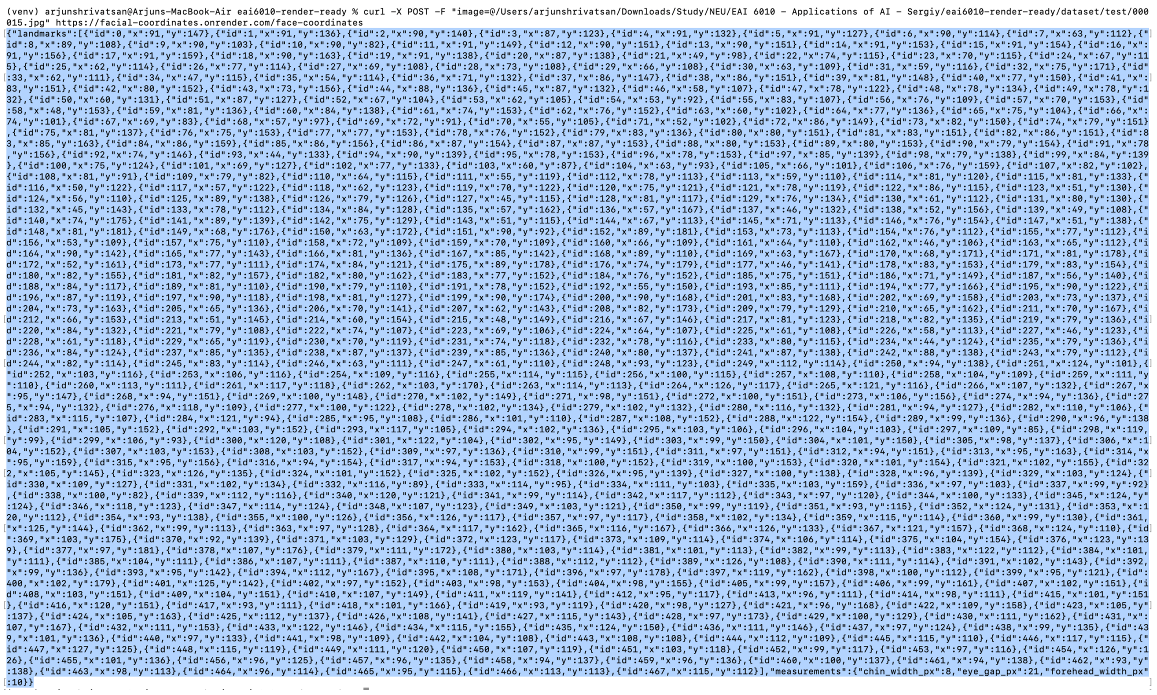
Keep the Render app active by making a request at intervals to prevent it from sleeping. If it goes to sleep, the first request might take a few seconds to wake it up. Ensure that large image files are resized for faster processing.

Running Instructions (Using CURL)

To invoke the facial landmark detection service, a `curl` command can be used to post an image to the `/face-coordinates` endpoint. The command follows the format:

```
bash
Copy code
curl -X POST -F "image=@/path/to/image.jpg" https://facial-coordinates.onrender.com/face-coordinates
```

e.g Shown image below :



```
(venv) arjunshrivatsan@Arjuns-MacBook-Air eai6010-render-ready % curl -X POST -F "image=@/Users/arjunshrivatsan/Downloads/Study/NEU/EAI 6010 - Applications of AI - Sergiy/eai6010-render-ready/dataset/test/000
015.jpg" https://facial-coordinates.onrender.com/face-coordinates
```

```
{"landmarks":[{"id":0,"x":91,"y":147},{"id":1,"x":91,"y":136},{"id":2,"x":90,"y":140},{"id":3,"x":87,"y":123},{"id":4,"x":91,"y":132},{"id":5,"x":91,"y":127},{"id":6,"x":90,"y":114},{"id":7,"x":63,"y":112},{"
id":8,"x":89,"y":108},{"id":9,"x":90,"y":103},{"id":10,"x":90,"y":82},{"id":11,"x":91,"y":149},{"id":12,"x":90,"y":151},{"id":13,"x":90,"y":151},{"id":14,"x":91,"y":153},{"id":15,"x":91,"y":154},{"id":16,"x":
91,"y":156},{"id":17,"x":91,"y":159},{"id":18,"x":90,"y":163},{"id":19,"x":91,"y":138},{"id":20,"x":87,"y":138},{"id":21,"x":49,"y":98},{"id":22,"x":74,"y":115},{"id":23,"x":70,"y":115},{"id":24,"x":67,"y":11
5},{"id":25,"x":62,"y":114},{"id":26,"x":77,"y":114},{"id":27,"x":69,"y":108},{"id":28,"x":73,"y":108},{"id":29,"x":66,"y":108},{"id":30,"x":63,"y":109},{"id":31,"x":59,"y":116},{"id":32,"x":75,"y":171},{"id"
:33,"x":62,"y":111},{"id":34,"x":47,"y":115},{"id":35,"x":54,"y":114},{"id":36,"x":71,"y":132},{"id":37,"x":86,"y":147},{"id":38,"x":86,"y":151},{"id":39,"x":81,"y":148},{"id":40,"x":77,"y":150},{"id":41,"x":
83,"y":151},{"id":42,"x":80,"y":152},{"id":43,"x":73,"y":156},{"id":44,"x":88,"y":136},{"id":45,"x":87,"y":132},{"id":46,"x":58,"y":107},{"id":47,"x":78,"y":122},{"id":48,"x":78,"y":134},{"id":49,"x":78,"y":1
32},{"id":50,"x":60,"y":131},{"id":51,"x":87,"y":127},{"id":52,"x":67,"y":104},{"id":53,"x":62,"y":105},{"id":54,"x":53,"y":92},{"id":55,"x":83,"y":107},{"id":56,"x":76,"y":109},{"id":57,"x":70,"y":153},{"id"
:58,"x":48,"y":153},{"id":59,"x":81,"y":136},{"id":60,"x":84,"y":138},{"id":61,"x":74,"y":153},{"id":62,"x":76,"y":152},{"id":63,"x":60,"y":102},{"id":64,"x":77,"y":136},{"id":65,"x":75,"y":104},{"id":66,"x":
74,"y":101},{"id":67,"x":69,"y":83},{"id":68,"x":57,"y":97},{"id":69,"x":72,"y":91},{"id":70,"x":55,"y":105},{"id":71,"x":52,"y":102},{"id":72,"x":86,"y":149},{"id":73,"x":82,"y":150},{"id":74,"x":79,"y":151}
,{"id":75,"x":81,"y":137},{"id":76,"x":75,"y":153},{"id":77,"x":77,"y":153},{"id":78,"x":76,"y":152},{"id":79,"x":83,"y":136},{"id":80,"x":80,"y":151},{"id":81,"x":83,"y":151},{"id":82,"x":86,"y":151},{"id":8
3,"x":85,"y":163},{"id":84,"x":86,"y":159},{"id":85,"x":86,"y":156},{"id":86,"x":87,"y":154},{"id":87,"x":87,"y":153},{"id":88,"x":80,"y":153},{"id":89,"x":80,"y":153},{"id":90,"x":79,"y":154},{"id":91,"x":78
,"y":156},{"id":92,"x":74,"y":146},{"id":93,"x":44,"y":133},{"id":94,"x":90,"y":139},{"id":95,"x":78,"y":153},{"id":96,"x":78,"y":153},{"id":97,"x":85,"y":139},{"id":98,"x":79,"y":138},{"id":99,"x":84,"y":139
},{"id":100,"x":75,"y":124},{"id":101,"x":69,"y":127},{"id":102,"x":77,"y":133},{"id":103,"x":60,"y":87},{"id":104,"x":63,"y":93},{"id":105,"x":66,"y":101},{"id":106,"x":76,"y":159},{"id":107,"x":82,"y":102},
{"id":108,"x":81,"y":91},{"id":109,"x":79,"y":82},{"id":110,"x":64,"y":115},{"id":111,"x":55,"y":119},{"id":112,"x":78,"y":113},{"id":113,"x":59,"y":110},{"id":114,"x":81,"y":120},{"id":115,"x":81,"y":133},{"
id":116,"x":50,"y":122},{"id":117,"x":57,"y":122},{"id":118,"x":62,"y":123},{"id":119,"x":70,"y":122},{"id":120,"x":75,"y":121},{"id":121,"x":78,"y":119},{"id":122,"x":86,"y":115},{"id":123,"x":51,"y":130},{"
id":124,"x":56,"y":110},{"id":125,"x":89,"y":138},{"id":126,"x":79,"y":126},{"id":127,"x":45,"y":115},{"id":128,"x":81,"y":117},{"id":129,"x":76,"y":134},{"id":130,"x":61,"y":112},{"id":131,"x":80,"y":130},{"
id":132,"x":45,"y":143},{"id":133,"x":78,"y":112},{"id":134,"x":84,"y":128},{"id":135,"x":57,"y":162},{"id":136,"x":57,"y":167},{"id":137,"x":46,"y":132},{"id":138,"x":52,"y":156},{"id":139,"x":49,"y":108},{"
id":140,"x":74,"y":175},{"id":141,"x":89,"y":139},{"id":142,"x":75,"y":129},{"id":143,"x":51,"y":115},{"id":144,"x":67,"y":113},{"id":145,"x":71,"y":113},{"id":146,"x":76,"y":154},{"id":147,"x":51,"y":138},{"
id":148,"x":81,"y":181},{"id":149,"x":68,"y":176},{"id":150,"x":63,"y":172},{"id":151,"x":90,"y":92},{"id":152,"x":89,"y":181},{"id":153,"x":73,"y":113},{"id":154,"x":76,"y":112},{"id":155,"x":77,"y":112},{"i
d":156,"x":53,"y":109},{"id":157,"x":75,"y":110},{"id":158,"x":72,"y":109},{"id":159,"x":70,"y":109},{"id":160,"x":66,"y":109},{"id":161,"x":64,"y":110},{"id":162,"x":46,"y":106},{"id":163,"x":65,"y":112},{"i
d":164,"x":90,"y":142},{"id":165,"x":77,"y":143},{"id":166,"x":81,"y":136},{"id":167,"x":85,"y":142},{"id":168,"x":89,"y":110},{"id":169,"x":63,"y":167},{"id":170,"x":68,"y":171},{"id":171,"x":81,"y":178},{"i
d":172,"x":52,"y":161},{"id":173,"x":77,"y":111},{"id":174,"x":84,"y":121},{"id":175,"x":89,"y":178},{"id":176,"x":74,"y":179},{"id":177,"x":46,"y":141},{"id":178,"x":83,"y":153},{"id":179,"x":83,"y":154},{"i
d":180,"x":82,"y":155},{"id":181,"x":82,"y":157},{"id":182,"x":80,"y":162},{"id":183,"x":77,"y":152},{"id":184,"x":76,"y":152},{"id":185,"x":75,"y":151},{"id":186,"x":71,"y":149},{"id":187,"x":56,"y":140},{"i
d":188,"x":84,"y":117},{"id":189,"x":81,"y":110},{"id":190,"x":79,"y":110},{"id":191,"x":78,"y":152},{"id":192,"x":55,"y":150},{"id":193,"x":85,"y":111},{"id":194,"x":77,"y":166},{"id":195,"x":90,"y":122},{"i
d":196,"x":87,"y":119},{"id":197,"x":90,"y":118},{"id":198,"x":81,"y":127},{"id":199,"x":90,"y":174},{"id":200,"x":90,"y":168},{"id":201,"x":83,"y":168},{"id":202,"x":69,"y":158},{"id":203,"x":73,"y":137},{"i
d":204,"x":73,"y":163},{"id":205,"x":65,"y":136},{"id":206,"x":70,"y":141},{"id":207,"x":62,"y":143},{"id":208,"x":82,"y":173},{"id":209,"x":79,"y":129},{"id":210,"x":65,"y":162},{"id":211,"x":70,"y":167},{"i
d":212,"x":66,"y":153},{"id":213,"x":51,"y":145},{"id":214,"x":60,"y":154},{"id":215,"x":48,"y":149},{"id":216,"x":67,"y":146},{"id":217,"x":81,"y":123},{"id":218,"x":82,"y":135},{"id":219,"x":79,"y":136},{"i
d":220,"x":84,"y":132},{"id":221,"x":79,"y":108},{"id":222,"x":74,"y":107},{"id":223,"x":69,"y":106},{"id":224,"x":64,"y":107},{"id":225,"x":61,"y":108},{"id":226,"x":58,"y":113},{"id":227,"x":46,"y":123},{"i
d":228,"x":61,"y":118},{"id":229,"x":65,"y":119},{"id":230,"x":70,"y":119},{"id":231,"x":74,"y":118},{"id":232,"x":78,"y":116},{"id":233,"x":80,"y":115},{"id":234,"x":44,"y":124},{"id":235,"x":79,"y":136},{"i
d":236,"x":84,"y":124},{"id":237,"x":85,"y":135},{"id":238,"x":87,"y":137},{"id":239,"x":85,"y":136},{"id":240,"x":80,"y":137},{"id":241,"x":87,"y":138},{"id":242,"x":88,"y":138},{"id":243,"x":79,"y":112},{"i
d":244,"x":82,"y":114},{"id":245,"x":83,"y":114},{"id":246,"x":63,"y":111},{"id":247,"x":61,"y":110},{"id":248,"x":93,"y":123},{"id":249,"x":112,"y":114},{"id":250,"x":94,"y":138},{"id":251,"x":124,"y":101},{
"id":252,"x":103,"y":116},{"id":253,"x":106,"y":116},{"id":254,"x":109,"y":116},{"id":255,"x":114,"y":115},{"id":256,"x":100,"y":115},{"id":257,"x":108,"y":110},{"id":258,"x":104,"y":109},{"id":259,"x":111,"y
":110},{"id":260,"x":113,"y":111},{"id":261,"x":117,"y":118},{"id":262,"x":103,"y":170},{"id":263,"x":114,"y":113},{"id":264,"x":126,"y":117},{"id":265,"x":121,"y":116},{"id":266,"x":107,"y":132},{"id":267,"x
":95,"y":147},{"id":268,"x":94,"y":151},{"id":269,"x":100,"y":148},{"id":270,"x":102,"y":149},{"id":271,"x":98,"y":151},{"id":272,"x":100,"y":151},{"id":273,"x":106,"y":156},{"id":274,"x":94,"y":136},{"id":27
5,"x":94,"y":132},{"id":276,"x":118,"y":109},{"id":277,"x":100,"y":122},{"id":278,"x":102,"y":134},{"id":279,"x":102,"y":132},{"id":280,"x":116,"y":132},{"id":281,"x":94,"y":127},{"id":282,"x":110,"y":106},{"
id":283,"x":115,"y":107},{"id":284,"x":121,"y":94},{"id":285,"x":95,"y":108},{"id":286,"x":101,"y":110},{"id":287,"x":108,"y":152},{"id":288,"x":122,"y":154},{"id":289,"x":99,"y":136},{"id":290,"x":96,"y":138
},{"id":291,"x":105,"y":152},{"id":292,"x":103,"y":152},{"id":293,"x":117,"y":105},{"id":294,"x":102,"y":136},{"id":295,"x":103,"y":106},{"id":296,"x":104,"y":103},{"id":297,"x":109,"y":85},{"id":298,"x":119,
"y":99},{"id":299,"x":106,"y":93},{"id":300,"x":120,"y":108},{"id":301,"x":122,"y":104},{"id":302,"x":95,"y":149},{"id":303,"x":99,"y":150},{"id":304,"x":101,"y":150},{"id":305,"x":98,"y":137},{"id":306,"x":1
04,"y":152},{"id":307,"x":103,"y":153},{"id":308,"x":103,"y":152},{"id":309,"x":97,"y":136},{"id":310,"x":99,"y":151},{"id":311,"x":97,"y":151},{"id":312,"x":94,"y":151},{"id":313,"x":95,"y":163},{"id":314,"x
":95,"y":159},{"id":315,"x":95,"y":156},{"id":316,"x":94,"y":154},{"id":317,"x":94,"y":153},{"id":318,"x":100,"y":152},{"id":319,"x":100,"y":153},{"id":320,"x":101,"y":154},{"id":321,"x":102,"y":155},{"id":32
2,"x":105,"y":145},{"id":323,"x":126,"y":135},{"id":324,"x":101,"y":152},{"id":325,"x":102,"y":152},{"id":326,"x":95,"y":139},{"id":327,"x":100,"y":138},{"id":328,"x":96,"y":139},{"id":329,"x":103,"y":124},{"
id":330,"x":109,"y":127},{"id":331,"x":102,"y":134},{"id":332,"x":116,"y":89},{"id":333,"x":114,"y":95},{"id":334,"x":111,"y":103},{"id":335,"x":103,"y":159},{"id":336,"x":97,"y":103},{"id":337,"x":99,"y":92}
,{"id":338,"x":100,"y":82},{"id":339,"x":112,"y":116},{"id":340,"x":120,"y":121},{"id":341,"x":99,"y":114},{"id":342,"x":117,"y":112},{"id":343,"x":97,"y":120},{"id":344,"x":100,"y":133},{"id":345,"x":124,"y"
:124},{"id":346,"x":118,"y":123},{"id":347,"x":114,"y":124},{"id":348,"x":107,"y":123},{"id":349,"x":103,"y":121},{"id":350,"x":99,"y":119},{"id":351,"x":93,"y":115},{"id":352,"x":124,"y":131},{"id":353,"x":1
20,"y":112},{"id":354,"x":93,"y":138},{"id":355,"x":100,"y":126},{"id":356,"x":126,"y":117},{"id":357,"x":97,"y":117},{"id":358,"x":102,"y":134},{"id":359,"x":115,"y":114},{"id":360,"x":99,"y":130},{"id":361,
"x":125,"y":144},{"id":362,"x":99,"y":113},{"id":363,"x":97,"y":128},{"id":364,"x":117,"y":162},{"id":365,"x":116,"y":167},{"id":366,"x":126,"y":133},{"id":367,"x":121,"y":157},{"id":368,"x":124,"y":110},{"id
":369,"x":103,"y":175},{"id":370,"x":92,"y":139},{"id":371,"x":103,"y":129},{"id":372,"x":123,"y":117},{"id":373,"x":109,"y":114},{"id":374,"x":106,"y":114},{"id":375,"x":104,"y":154},{"id":376,"x":123,"y":13
9},{"id":377,"x":97,"y":181},{"id":378,"x":107,"y":176},{"id":379,"x":111,"y":172},{"id":380,"x":103,"y":114},{"id":381,"x":101,"y":113},{"id":382,"x":99,"y":113},{"id":383,"x":122,"y":112},{"id":384,"x":101,
"y":111},{"id":385,"x":104,"y":111},{"id":386,"x":107,"y":111},{"id":387,"x":110,"y":111},{"id":388,"x":112,"y":112},{"id":389,"x":126,"y":108},{"id":390,"x":111,"y":114},{"id":391,"x":102,"y":143},{"id":392,
"x":99,"y":136},{"id":393,"x":95,"y":142},{"id":394,"x":112,"y":167},{"id":395,"x":108,"y":171},{"id":396,"x":97,"y":178},{"id":397,"x":119,"y":162},{"id":398,"x":100,"y":112},{"id":399,"x":95,"y":121},{"id":
400,"x":102,"y":179},{"id":401,"x":125,"y":142},{"id":402,"x":97,"y":152},{"id":403,"x":98,"y":153},{"id":404,"x":98,"y":155},{"id":405,"x":99,"y":157},{"id":406,"x":99,"y":161},{"id":407,"x":102,"y":151},{"i
d":408,"x":103,"y":151},{"id":409,"x":104,"y":151},{"id":410,"x":107,"y":149},{"id":411,"x":119,"y":141},{"id":412,"x":95,"y":117},{"id":413,"x":96,"y":111},{"id":414,"x":98,"y":111},{"id":415,"x":101,"y":151
},{"id":416,"x":120,"y":151},{"id":417,"x":93,"y":111},{"id":418,"x":101,"y":166},{"id":419,"x":93,"y":119},{"id":420,"x":98,"y":127},{"id":421,"x":96,"y":168},{"id":422,"x":109,"y":158},{"id":423,"x":105,"y"
:137},{"id":424,"x":105,"y":163},{"id":425,"x":112,"y":137},{"id":426,"x":108,"y":141},{"id":427,"x":115,"y":143},{"id":428,"x":97,"y":173},{"id":429,"x":100,"y":129},{"id":430,"x":111,"y":162},{"id":431,"x":
107,"y":167},{"id":432,"x":111,"y":153},{"id":433,"x":122,"y":146},{"id":434,"x":115,"y":155},{"id":435,"x":124,"y":150},{"id":436,"x":111,"y":146},{"id":437,"x":97,"y":124},{"id":438,"x":99,"y":135},{"id":43
9,"x":101,"y":136},{"id":440,"x":97,"y":133},{"id":441,"x":98,"y":109},{"id":442,"x":104,"y":108},{"id":443,"x":108,"y":108},{"id":444,"x":112,"y":109},{"id":445,"x":115,"y":110},{"id":446,"x":117,"y":115},{"
id":447,"x":127,"y":125},{"id":448,"x":115,"y":119},{"id":449,"x":111,"y":120},{"id":450,"x":107,"y":119},{"id":451,"x":103,"y":118},{"id":452,"x":99,"y":117},{"id":453,"x":97,"y":116},{"id":454,"x":126,"y":1
26},{"id":455,"x":101,"y":136},{"id":456,"x":96,"y":125},{"id":457,"x":96,"y":135},{"id":458,"x":94,"y":137},{"id":459,"x":96,"y":136},{"id":460,"x":100,"y":137},{"id":461,"x":94,"y":138},{"id":462,"x":93,"y"
:138},{"id":463,"x":98,"y":113},{"id":464,"x":96,"y":114},{"id":465,"x":95,"y":115},{"id":466,"x":113,"y":113},{"id":467,"x":115,"y":112}],"measurements":{"chin_width_px":8,"eye_gap_px":21,"forehead_width_px"
:10}}
```

This allows easy testing from the command line, especially for developers or instructors evaluating the microservice. The key part of the command is `-F "image=@/path/to/image.jpg"`, which uploads the image as form data under the key `image`.

This method is simple, doesn't require a frontend interface, and works well with most Linux, macOS, and Windows environments that have cURL installed. No additional headers or tokens are required, making the service frictionless to test.

Conclusion

This project demonstrates how AI can be meaningfully applied through a lightweight and accessible microservice. By leveraging MediaPipe for facial landmark detection (Lugaresi et al., 2019; Google Developers, 2023), we enabled efficient face geometry analysis without heavy computation. The CelebA dataset (Liu et al., 2015) served as a robust testbed, providing facial diversity and alignment quality. Deployment via Render.com ensured public accessibility and reproducibility (Render, 2024). The use of `curl` allowed straightforward command-line testing and integration. This implementation proves the potential of real-time facial geometry analysis in applications such as custom-fit helmet design and safety gear innovation.

References

Google Developers. (2023). *MediaPipe face mesh*. https://developers.google.com/mediapipe/solutions/vision/face_mesh

Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 3730–3738).

Lugaresi, C., Tang, J., Nash, H., McGuire, M., Tseng, Y., Chang, C., ... & Kwatra, V. (2019). *MediaPipe: A framework for building perception pipelines* (arXiv:1906.08172). arXiv. https://arxiv.org/abs/1906.08172

OpenCV Team. (2023). *OpenCV: Open source computer vision library*. https://opencv.org/

Render. (2024). *Render.com documentation: Web services*. https://render.com/docs/web-services

Zafeiriou, S., Trigeorgis, G., Chrysos, G. G., Deng, J., & Tzimiropoulos, G. (2017). The Menpo facial landmark localisation challenge: A step towards the solution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 2116–2125). https://openaccess.thecvf.com/content_cvpr_2017_workshops/w5/html/Zafeiriou_The_Menpo_Facial_CVPR_2017_paper.html