DEVELOPING A DESIGN DECISION-SUPPORT SYSTEM DESIGN REPORT

'HANDI': A Hand Contour Finder for All Abilities – An Adaptive Tool for Custom Glove Making

Author. Nina Jiaying Lai, z5319371, UNSW
Couse. CODE2120, Nov 2024
Course Convenor. Daniel Yu

CONTENTS

INTRODUCTION & BACKGROUND 03

RESEARCH & ANALYSIS 04
- INTEROPERABILITY STUDY.
- IDENTIFY THE SYSTEM, HOW DATA FORMAT, PROTOCOLS AND CHANLEGES
- EXAMINE THE APPLICATION IDEA

API PROPOSAL 06

SYSTEM DIAGRAM 07

DESIGN DEVELOPMENT 08
- TESTING METHODOLOGY.
- ANALYSIS OF RESULTS.
- ITERATIVE REFINEMENT.

DISCUSSION 12
- CONTRIBUTION TO UN SDGS.

CONCLUSION 12

REFERENCE 14
- A LIST OF FIGURES

APPENDIX 15
- VIDEO DEMONSTRATION
- README INSTRUCTION
- PRINT TEMPLATE
- WEEKLY MEETING RESULTS

INTRODUCTION

BACKGROUND STUDY.

OPTIMIZING THE PRODUCTION OF HAND PRODUCTS/PLAIN GLOVES THROUGH API MODIFICATIONS

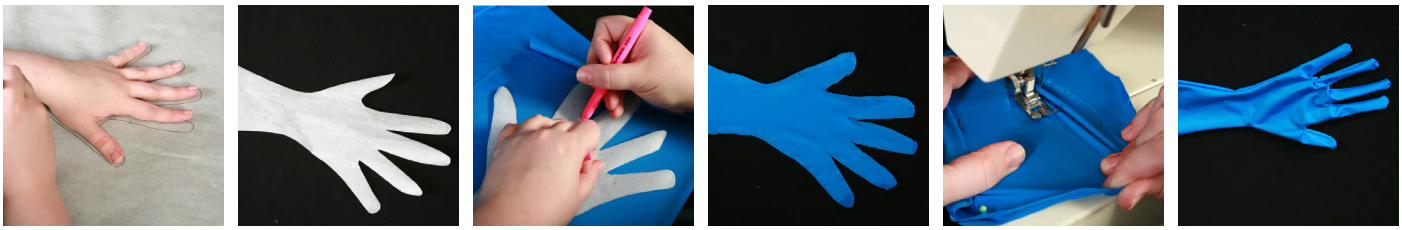


FIGURE 1 TO 6. TRADITIONAL PROCESS OF MAKING PLAIN SPANDEX GLOVES

15-20 MINUTES PER GLOVE

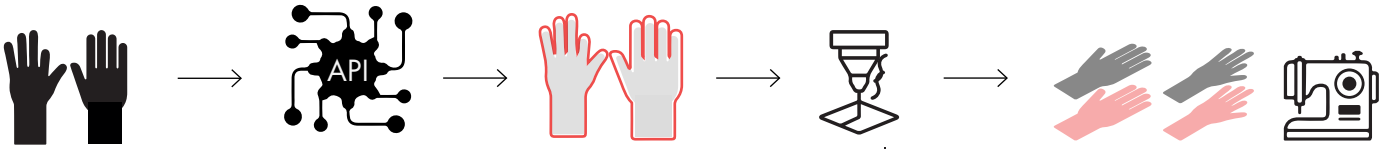


FIGURE 7. OPTIMIZED DESIGN PROPOSAL FOR THE FABRICATION SYSTEM

10-20 SECONDS PER GLOVE

In the rapidly developing fields of computational design and industrial design, customization and accessibility are two important elements of user experience design. Designing items that cater to individual requirements—particularly for users with physical differences—this presents a design opportunity to merge human compassion with cutting-edge technology. This project, Customized Hand Models and Adaptive Products for All Abilities, aims to design a caring and useful application by generating vector documents from input images for the production of fully customizable hand products, such as gloves, depend on user-specific hand outlines and dimensions.

By incorporating data collections, streamlined workflow, parametric design, and advanced fabrication techniques, the initiative seeks to address a significant gap in the AEC(Architecture, Engineering and Construction) market of industrial products for humans : adaptive solutions for individuals with unique hand attributes, such as missing or incomplete fingers. These caring designs guarantee that no one is left behind, so those custom-fit plain gloves enhances both functionality and comfort.

A basic plain glove are made by six necessary stages (FIGURE 1 to 6). The initial stages for preparing the refernce for hands with fingers open require tedius manul work. At the same time, the craftsman also needs to hold the pen perpendicular to the tracing paper by using another hand to ensure the accuracy and consistenct of the ‘pattern’, suggested from Marie (2021). This process is extremely user-unfridenly and could be improved from API support.

In contrast, the API revolutionizes the way people get the template and accelerates the manufacturing through the intergration of laser cutting (FIGURE 7). Consequently, the enhanced proposal saves time and providing advantages for large scale production.

Application Idea: A Hand Contour Finder for All Abilities

To support designers to create hand outlines in manufacturing hand products such as gloves, based on images of individual hands, with an emphasis on adaptability and personalization

Data Collection Strategy:

- Identify Data Needs: TheAPI needs data from images of users hands, including hand scale, dimensions (finger lengths, plam sizes), clear hand outlines (standing out from background), as well as any missing or incomplete features

Source Data:

- User Images:
  - Pictures of user hands are uploaded to use as references to extract and analyze the hand contours and dimentions
  - Qrcode should be sticked to users hand as part of scale reference

Access Data Quality:

- Cross-verify the hand dimensions from more than one images sources (e.g., different lightning conditions or angles) to guarantee the reliability in the vector generation

Data Integration Plan:

- Roboflow Integration:
  - Roboflow is utilized to preprocess and augment the image dataset, improving the accuracy in the image detection by training model with a vast arrays of hand images, such as starting with a smaller dataset (100 images)
  - Organize data into a relational database for efficient quering. In this case, the API only stores the vector file to be further design and fabrication process

Application:

- Contour Vector Output:
  - The output result of the API is the hand contour vector (the outline), which will be used by industrial designers to create custom products, such as gloves, sourcing from the specific hand shapes and dimension from users

ANALYSIS

INTEROPERABILITY STUDY.

IDENTIFY THE SYSTEM, HOW DATA FORMAT, PROTOCOLS AND CHANLEGES

EXAMINE THE APPLICATION IDEA

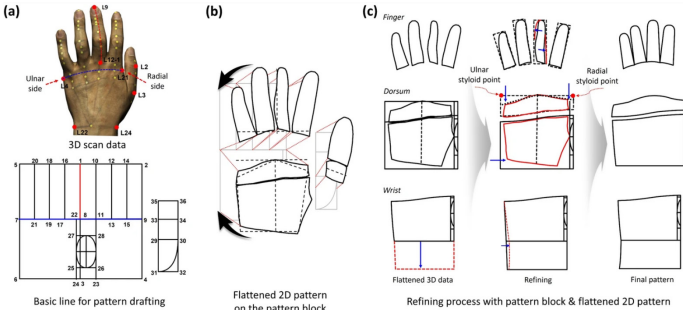


FIGURE 8. GLOVE PATTERN DRAFTING METHOD: (A) 3D SCAN DATA AND PATTERN BLOCK, (B) PATTERN DRAFTING USING FLATTENED 3D SHAPE AND PATTERN BLOCK, AND (C) REFINING PROCESS

EXISTING CASE STUDY 01

The relevant interest has accruing in hand-type assistive products from medical perspective for finger mobility. Lee and Park (2024) developed an ergonomic methodology in drafting the pattern based on 3D scan data, which helps to create bespoke glove (FIGURE 8). However, this process was still manual based at the end.

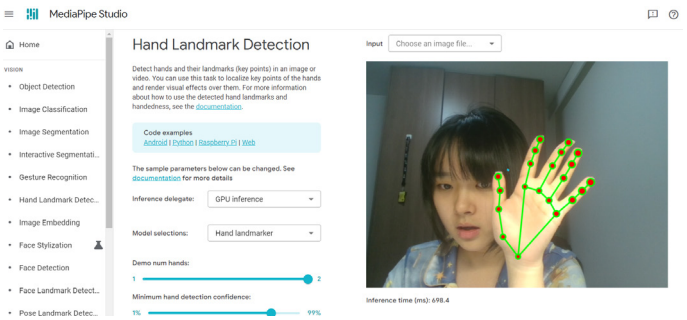


FIGURE 9. HAND LANDMARKS DETECTION BY GOOGLE

EXISTING CASE STUDY 02

This API detects for real-time hand landmarks and keypoints in the video streams, which is commonly seen in gesture recognition. The app detects 21 keypoints on each hand including joints and palm points (FIGURE 9). In terms of the detection for contours, this method could possibly enhance precision.

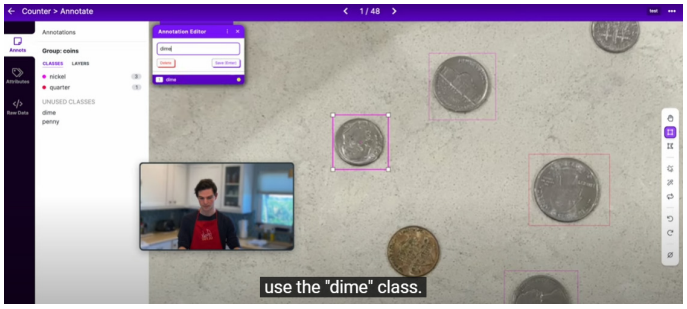


FIGURE 10. COIN DETECTION AND LABELING

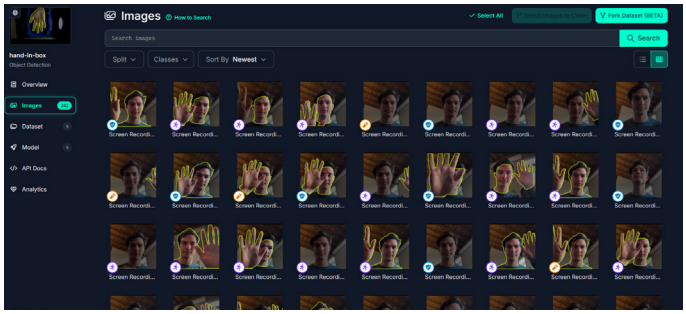


FIGURE 11. HAND IN BOX

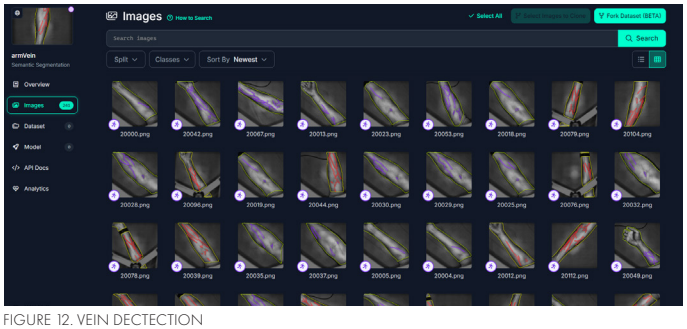


FIGURE 12. VEIN DECTECTION

EXISTING CASE STUDY 03

The coin detection model is trained under a great number of pictures by manually labelling the coins with a box shape, to distinguish the types (FIGURE 10). A more complicated tracing method could be found in vein and hand detection samples. Nevertheless, none of these experiments indicate the scale the size of the detect object (FIGURE 11&12).

Objective: What Problem Does Your API Solve?

The API revolutionizes the way of tracing hand outlines as the design need in industries such as for glove manufacturing, medical prosthetics, and custom product design. It allows craftspeople to upload images of hands, which are processed to capture precise hand contours. By using QR code as a reference point, the app enables to gather the information of dimensions.

Data Types:

- Hand Image (e.g. User uploads a hand image, including QR on plam for scaling)
- Contours & Dimensions: provides hand outlines defined by points (lines) enabling users to measure the true dimensions of their hands in the real-world.
- Others: The user ID, and time could be inserted to the platform

Methods:

- GET Requests: get a corresponded hand outline data
- POST Request: upload images for detection and dimension calculation
- DELETE Requests: update or delete the history

Endpoints:

- /upload_image
- /get_contour/ {id}
- /calculate_size/ {id}
- /projects{id}/models

Innovation and Feasibility:

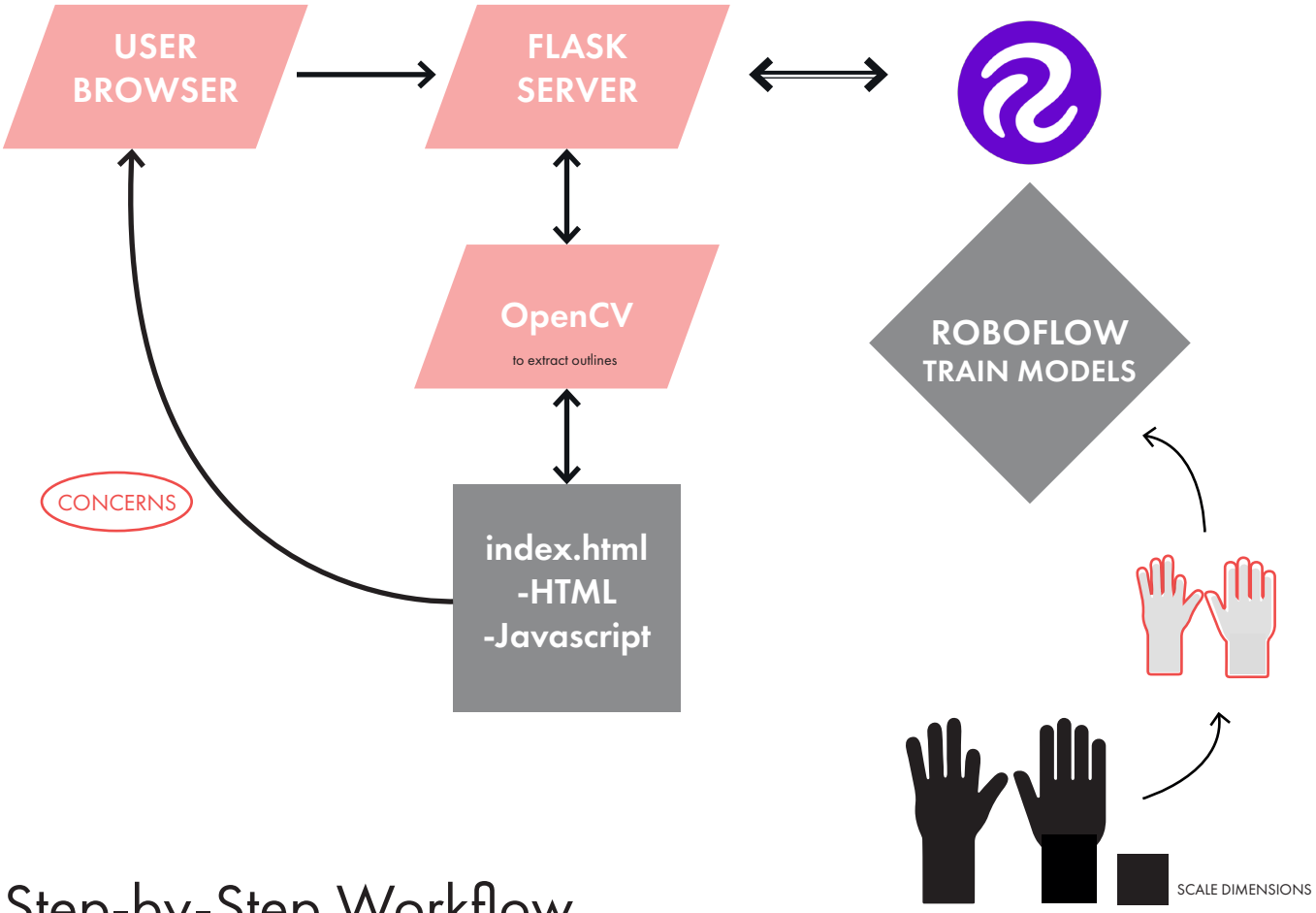
- Uniqueness
  - This is a caring design with potential medical development for prosthetic and no one is left behind. It helps the development of adaptive design in a new way which make this project different from other hand detection APIs in the market. The real-world QR code dimensions is design to be based on a fixed-size for ratio configurations to get the exact dimension.

Scalebility and Feasibility:

- This app could be easily scale up by using Google Cloud or other database. In terms of the technical concerns, it is based on standard protocols such as HTTP, and other smart technology plastforms like Roboflow for image contouring, and OpenCV for image processing.
- Concerns: Imagine if the image quality varies—how will the system perform consistently?
  - Solution 1: Establish a perfect testing environment; however, this may lack accessibility.
  - Solution 2: Use the alignment of the QR code as a reference plane to ensure the correct angle is detected.
  - Solution 3: Asking for taking multiple photos

SYSTEM DIAGRAM

EXPLAIN.
HOW THE API/APP WILL FUNCTION
AND FRONTEND COMPONENTS



Step-by-Step Workflow

1. Data Collection
2. Model Training in Roboflow
3. Python Setup
4. API Development:
   - POST Request: Allow users to upload hand and QR code
   - Use Roboflow Model: detect the hand and QR code
   - Extract Hand Contours: After detection, use OpenCV to extract the hand contour
   - Calculate Real Dimention
   - Return Results: Send back the hand contour and dimensions as a JSON response.
5. Testing and Optimization the API

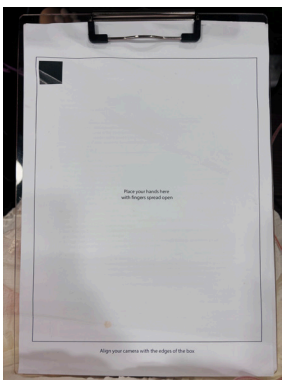


FIGURE 13. PRINT TEMPLATE TO FIND USERS

DESIGN DEVELOPMENT

TESTING METHODOLOGY.

HOW TO COLLECT DATA AND TRAIN IN ROBOFLOW

Using Roboflow Inference, YOLOv8 Instance Segmentation model can be deployed to train individual datasets, which is a different portal compared to Object Detection. The project was stuck in the middle of the process in deploying in the catefory of object detection (FIGURE 13).

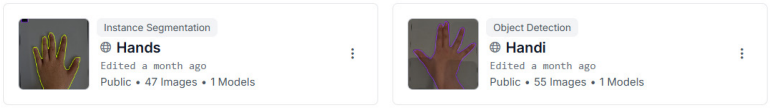


FIGURE 14. INSTANCE SEGMENTATION AND OBJECT DETECTION

47 images were successfully trained into the dataset with detailed annotation drawing for hands and squares (FIGURE 14). Specifically, in case to ensure the adaptability of this API, there are images to mimic different conditions for all users, such as only the fist, or Syndactyly for whom are born with fused or webbed fingers.

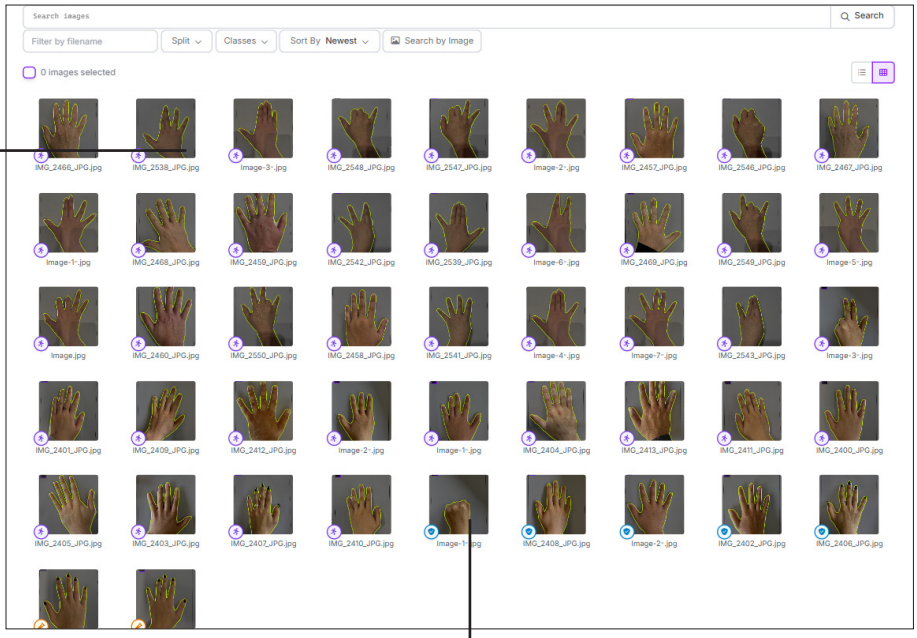
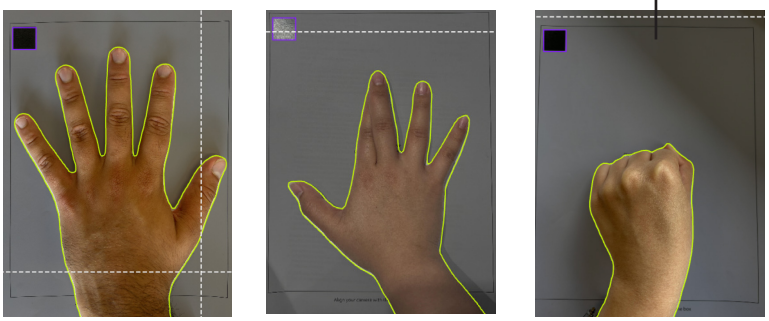


FIGURE 15. DATASET IN ROBOFLOW

Overall, 47 images with each annotated with an average of 2 annotations, results in a total of 94 annotations spread across 2 classes (FIGURE 15). The resolutaion is relatively high for each with a standard scale of 24.47 megapixels in a portrait orientation. 4284x5712 is the median image ratio.



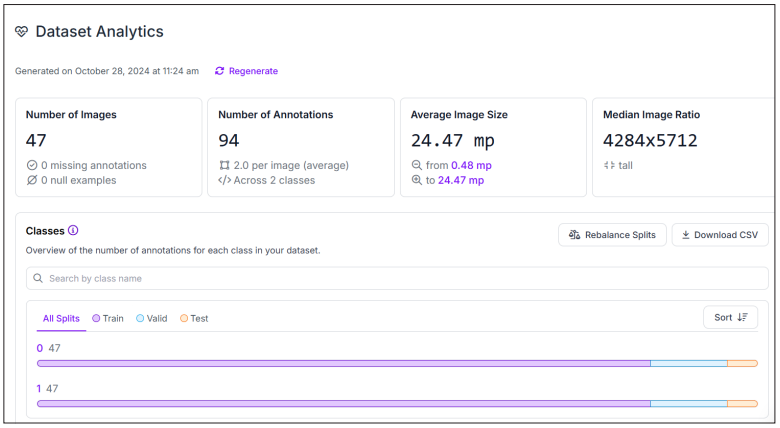


FIGURE 16. DATA ANALYTICS

DESIGN DEVELOPMENT

ANALYSIS OF RESULTS
TO HIGHLIGHT STRENGTHS, WEAKNESSES
AND USER EXPERIENCE

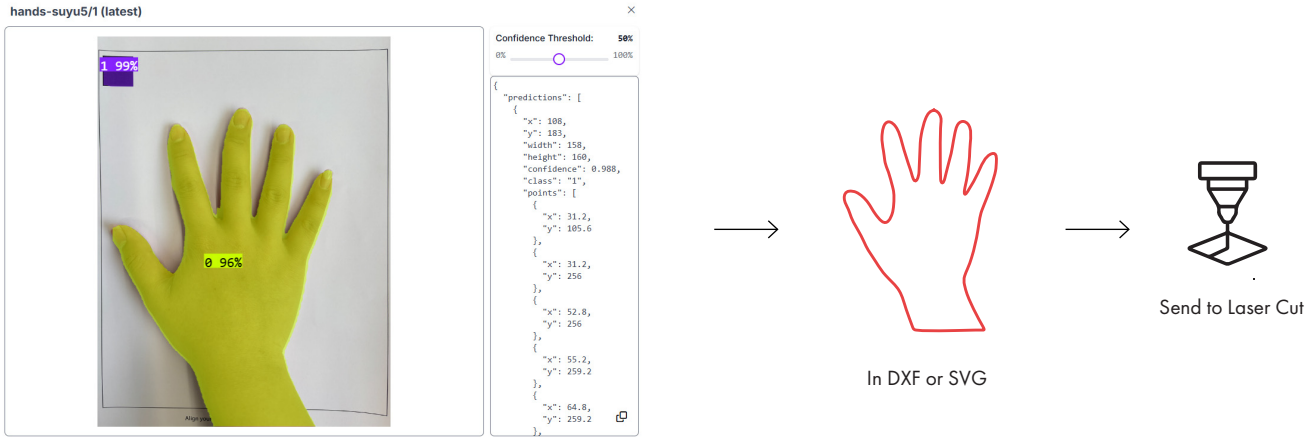


FIGURE 17. SUCCEED IN ROBOFLOW

Strengths of the Dataset

- Balanced Classes: Equal representation of both classes ensures fairness and avoids skewed model predictions.
- High Resolution: High-resolution images capture fine details, useful for tasks like object detection or fine-grained classification.
- Finished Data: No missing annotations or null examples, making it ready for model training without preprocessing issues.

Weakness of the Dataset

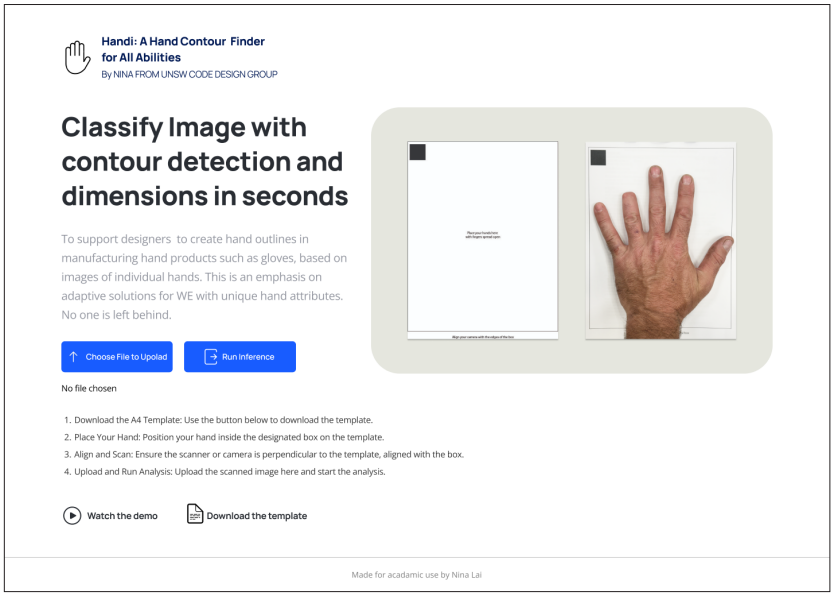
- High Resolution: Large image can lead to high memory and computational requirements during training.
  Suggested preprocessing: Resize images smaller for faster training.
- Annotation Density: may have sparse annotation coverage.
- Small Dataset: less than 50 images, which might not be sufficient for training a robust machine learning model.

TESTING METHODOLOGY		
Step	**Description**	**Technologies/Outputs**
Training and Deployment	Train a machine learning model using RoboFlow. Deploy it via Flask for integration with the web framework and API.	RoboFlow, Flask, Pre-trained ML Model
Upload Image	Users upload high-resolution hand images via the web interface. Supported formats include PNG and JPEG.	Web Interface (Flask), High-Resolution Image Support
Run Inference	The uploaded image is processed using the RoboFlow library or a local model to detect hand contours and annotations.	RoboFlow Library, Flask API
Get the Result	The processed results are returned, including segmented hand contours and confidence scores, visible in the web interface.	Web Interface Display, JSON API Response
Save as SVG	Save the contours in SVG format for editing in design tools (e.g., Illustrator) or export as DXF for laser cutting.	SVG for Design Tools (Illustrator, Rhino), DXF for Laser Cutting

DESIGN DEVELOPMENT

ANALYSIS OF RESULTS

UI/UX INTERFACE DESIGN AND USER EXPERIENCE



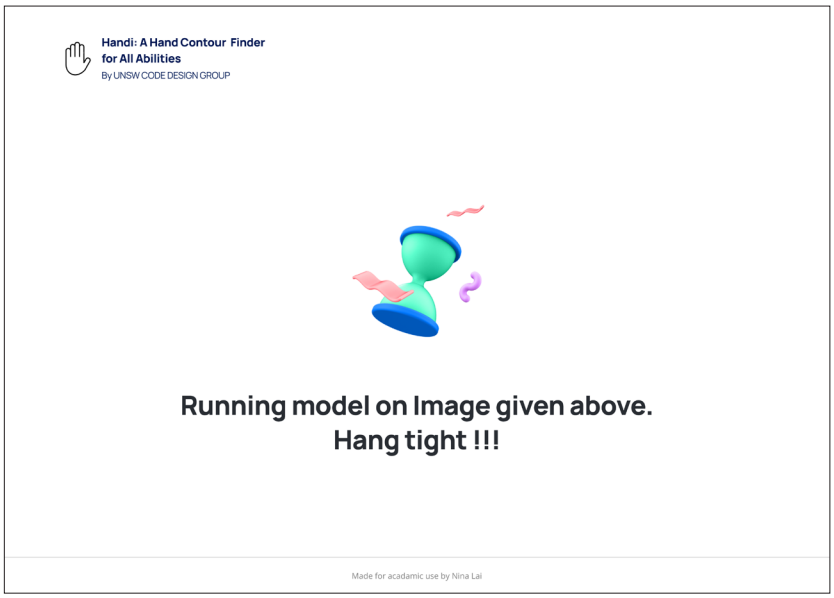
This main page as shown in FIGURE 18, is designed to clealy introduce users the project and enables users to upload images and follow instructions without overwhelming information.

Key Code Snippets

```
<form action="/upload" method="post" enctype="multipart/form-data">
  <input type="file" name="file" id="file-upload" required style="display: none;">
  <label for="file-upload" class="button">Choose File to Upload</label>
  <div id="file-chosen">No file chosen</div>
  <button type="submit" class="button">Run Inference</button>
</form>

const fileInput = document.getElementById("file-upload");
const fileChosen = document.getElementById("file-chosen");
fileInput.addEventListener("change", () => {
  fileChosen.textContent = fileInput.files[0]?.name || "No file chosen";
});
```

FIGURE 18. HOMEPAGE

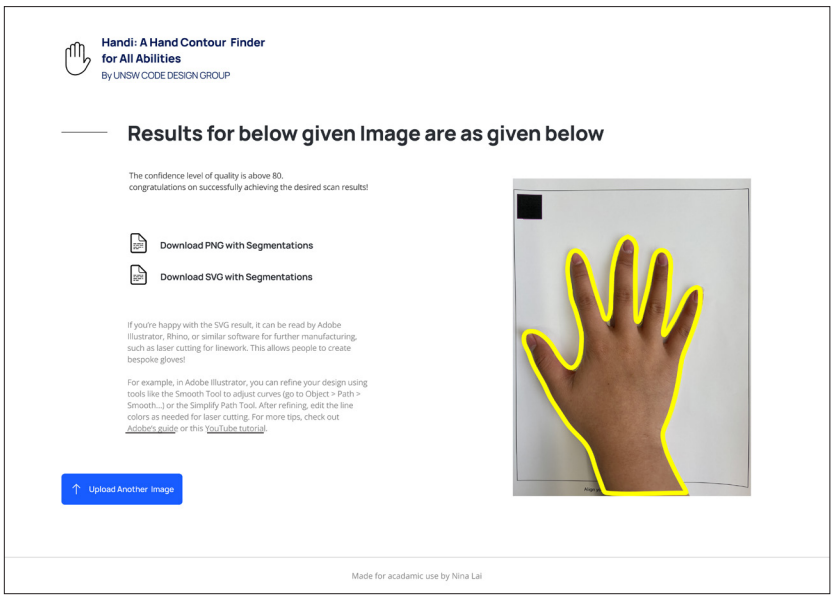


A loading page is important to remind people to hang on tight to wait the data is being processed, such as an animation with a rotating image.

Key Code Snippets

```
window.onload = () => {
  fetch("/start-process", { method: "POST" })
    .then(response => response.json())
    .then(data => {
      if (data.success) {
        window.location.href = data.redirect;
      } else {
        alert("Processing failed: " + data.error);
      }
    })
    .catch(error => alert("An error occurred: " + error));
};
```

FIGURE 19. PROCESSING PAGE



As a matter of fact, there is a need to value the uploading information. The quality of result might be varied due to how the uploaded picture is taken, like the lighting and the camera angles.

Key Code Snippets

```
<script>
  // Determine feedback based on confidence level
  if (confidenceLevel < 80) {
    document.getElementById("feedback").textContent =
      "The confidence level is below 80%. Please retake the picture for better results.";
    document.getElementById("feedback").style.color = "red"; // Highlight in red
  } else {
    document.getElementById("feedback").textContent =
      "The confidence level is above 80%. Great job achieving the desired scan
results!";
    document.getElementById("feedback").style.color = "green"; // Highlight in green
  }
</script>
```

FIGURE 20. RESULT PAGE

DESIGN DEVELOPMENT

ITERATIVE REFINEMENTS
SEGMENTATION QUALITY
AND SCREENSHOTS

FIGURE 21. REFINEMENT OF RESULTS

The great development of 'HANDI'involves three key steps. Each stage contributes to the overall efficiency and accuracy. Initialliy, the attention was on preprocessing the default settings, such as resizing and adjusting recoloring to standardize all pictures.

A considerable amount of effort is taken to test the edge detection and thredholding techniques. In the second step, challenges emerged to ensure the two colors to trace exactly for hand and square. Sometimes, the API results in the same color. Lastly, there is a thorny issue of turning curves to clean polygons to avoid hectic and overly busy line works in the upper half of FIGURE 21. In the review of Handi's performance, the model runs at an average generation speed of 11 seconds, and it eliminates the waiting times with seamless functionality of uploading and downloading. This process highlights the API great capability in time efficiency, to create bespoke hand gloves and contours. Hence, Handi brings about flexbility and readiness for manufacturing.

DISCUSSION

CONTRIBUTIONS TO UN SDG GOALS AND AREAS FOR SUSTAINABILITY

The Handi API showcases how innovative design strategy and advanced technological solutions in relate to inclusivity and user care. Also, this design approach contributes directly to Sustainable Development Goals (SDGs) 9, 10, anad 12 in fostering sustainable industrial implementations, reducing inequalities and promoting responsible production.

The real time detection guarantees the production for making bespoke hand gloves is precise and adaptable. With the downloaded svg, it exemplifies the potential in gcode or other landmark location file, which helps to layout optimization to save material waste when hundreds of gloves need to be manufactured. Compared to the traditional way of tracing hands, the project eliminates the manual work as stated at the begining of this study. To free people from toil and drugery, the integration of industrial design and computational design workflow revolutionizes and supports the infrastructure of smart production systems.

No one is left behind in this scenario. The idea embodies the heart of SDG 10, reducing disparties and bridge marginalized group.

Surprisingly, auto sewing laser cutting machines have been already put into practices. Thus, by using such advanced technologies and recyclable materials, the project is likely to optimize resource usage and succeed in responsible consumption.

CONCLUSION

In general, the majority of existed APIs are able to detect hands with exact landmark locations and operates in real time while users open the camera. However, there is a lack of integration of applying this detection to product applications, in which how to control the size dimension and extract contour lines turns into the design challenges that worth considering. From the proposal for this API, it concerns for all abilities and adapts the empathy approach to the needs of the ignored market. By developing OpenCV, Roboflow, and the fixed size of qrcode, designers are allowed to upload pictures and get the outline result at the end. It is interesting to see how fine tuned and accurate that the api could bring.

The Handi project suceed in refining the user journey and interface design with explicit studies and testings, though the detection is trained and relies heavily on Roboflow. However, some solutions worth reconsidering. For instance, without implementing direction landmark-located data in G-code files, it limits the potential for direct production applications. In addition, the failure to deploy the system in Pycharm is a great pity, to broaden the usability in a real time running website.

Acknowledgements.

I would like to express my gratitude to my tutor, Daniel Yu, for his support and great guidance throughout the development of this project.

My sincere thanks also go to all the volunteers and my classmates in CODE2021 for generously contributing hand images to support this project.

REFERENCE

Lee, S., & Park, J. (2024). Ergonomic glove pattern drafting method for hand assistive devices: considering 3D hand dimensions and finger mobility. *Fashion and Textiles, 11*(1). https://doi.org/10.1186/s40691-024-00397-5

Marie. (2021, January 16). *How to make basic spandex gloves.* Spandex Simplified. https://spandexsimplified.com/how-to-make-basic-spandex-gloves/

LIST OF FIGURES

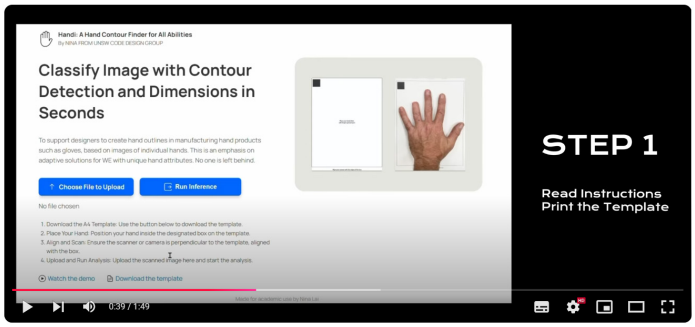
FIGURE 1 to 6
Spandex Simplified. (2021). *How to make basic spandex gloves.* Retrieved March 2024, from https://spandexsimplified.com/how-to-make-basic-spandex-gloves/

FIGURE 7
My Illustrations of Optimized Fabrication Process

FIGURE 8
Lee, S., & Park, J. (2024). Ergonomic glove pattern drafting method for hand assistive devices: considering 3D hand dimensions and finger mobility. *Fashion and Textiles, 11*(1). https://doi.org/10.1186/s40691-024-00397-5

FIGURE 9 TO 21
My Screenshots

VIDEO DEMO



Handi API Demonstration_CODE2120

YouTube

https://www.youtube.com/watch?v=-ENTk9LC_BY

README INSTRUCTION

Handi: A Hand Contour Finder for All Abilities

Instructions for Downloading and Setting Up the Project

Thank you for taking the time to review this project! Below are the steps to download, set up, and launch the API and application for grading:

1. Download the Project Files
1. Locate the provided **ZIP file** containing the project files.
2. Download the ZIP file to your computer.
3. Extract the contents of the ZIP file into a dedicated folder on your system.

2. Prerequisites
Before proceeding, ensure the following are installed on your system:
- **Python 3.7 or higher**: [Download Python](https://www.python.org/downloads/)
- **pip**: The Python package manager (included with most Python installations).
- **An IDE or code editor** (e.g., VS Code, PyCharm, or any terminal-based environment).

3. Install Required Python Libraries
1. Open a terminal or command prompt.
2. Navigate to the folder where you extracted the project files. For example:
   ```bash
   cd path/to/extracted/project-folder
   ```
3. Install the required Python dependencies by running:
   ```bash
   pip install -r requirements.txt
   ```

   This command will install all necessary libraries for the project, including:
   - **Flask**: For building the API.
   - **OpenCV**: For image processing tasks.
   - **RoboFlow**: For integrating pre-trained models (if applicable).
   - **Flask-CORS**: For managing cross-origin requests.
   - Additional libraries required for the project.

4. Prepare Additional Resources (If Required)
- **RoboFlow API Key**: If the application integrates with RoboFlow, ensure you have a valid API key. This can be placed in a `.env` file or a project configuration file as indicated in the code/documentation.
- **Pretrained Models**: If the project requires pretrained machine learning models, ensure they are downloaded or configured according to the provided instructions.

5. Launching the API
1. In the terminal, ensure you are still in the project directory.
2. Start the API server by running:
   ```bash
   python app.py
   ```

   Once the server starts, you will see output in the terminal indicating the server is running. By default, the API will be accessible at:
   ```
   http://127.0.0.1:5000/
   ```

   You can access this endpoint via:
   - Your browser (for testing purposes).
   - Tools like **Postman** for uploading and testing images.

6. Using the Application
If the project includes a frontend interface:
1. Open your browser and visit:
   ```
   http://127.0.0.1:5000/
   ```
2. Follow the on-screen instructions to upload images and process them using the application.

7. Testing the API
1. To test the API without the web interface:
   - Use a tool like **Postman** or **cURL** to send a POST request to the endpoint:
     ```
     http://127.0.0.1:5000/upload
     ```
   - Attach an image file in the request body.
   - Review the response in the tool or terminal.

   Example `cURL` command:
   ```bash
   curl -X POST -F "file=@your_image.png" http://127.0.0.1:5000/upload
   ```

8. Notes for Grading
- **File Structure**: The project files include a README, `requirements.txt`, and all the necessary scripts for easy setup.
- **Code Documentation**: Comments and documentation are included in the code to explain its functionality.
- **Results Validation**: Uploaded images will generate outputs that can be downloaded and inspected via the interface or API.

9. Troubleshooting
- **Missing Dependencies**: Ensure `pip install -r requirements.txt` completes without errors. Update `pip` if necessary using:
  ```bash
  pip install --upgrade pip
  ```
- **CORS Issues**: If there are cross-origin errors, confirm that the `Flask-CORS` library is installed and properly configured.
- **Environment Variables**: If the application relies on external APIs like RoboFlow, verify the `.env` file or configuration includes the correct keys.

Feel free to reach out if there are any issues or additional requirements for grading. Thank you for reviewing this project!

Place your hands here
with fingers spread open

Align your camera with the edges of the box

APPENDIX

WEEKLY MEETING AND PROGRESS