Stain Classification & Washing Recommendation App, Website
StainAway : *Snap, Spot, Solve*

G27
Team Members:
Yehseul Shin- 2025097965/ Junwoo Lee –2025045014/ Yongjun Lee – 2025075132
Yeongchan Ju – 2025065541/ A Jung KIm–2025033063

1. Introduction

1.1 Background

Stains on clothing are a common problem in daily life, especially when we eat food, beverages or working and exercising. However, many people are struggling with how to remove the stain in the proper way that fits with a specific stain. Since the laundry method is different by stain's type, applying the wrong method can not remove stain or cause damage to the fabric.
Although there is various laundry information on the internet, we considered users who got stains without realizing it and who don't have laundry knowledge so just washing with soap. As a result, users rely on guesswork, which leads to dirty clothes, waste of time and laundry resources and frustration.
Based on this situation, we recognized the need for a simple and accessible way to help users identify stains before washing, and to provide easy, reliable advice on how to clean them.
If users can identify stains at an early stage, it can reduce fabric damage, unnecessary washing and also can increase reliability for the users who want to know specialized stain removal methods.

1.2 User persona

Name: Yehseul Shin
Age: 19
Occupation: University freshman
Living situation: Dormitory
Background
As a freshman, she lived in a dormitory for the first time, away from my parents and family. Because she often spilled food and drinks, she frequently had to remove stains from my clothes. Unlike at home, where her parents handled laundry, stains wouldn't come out easily with simple soap or the washing machine. In addition, unexpected stains such as dirt, ink or unknown marks sometimes appeared on her clothes without a clear cause.
Pain points

- Difficulty identifying what kind of stain on the clothing.
- Applying incorrect washing methods causes dirty clothes.
- Possible fabric damage due to inappropriate cleaning methods.
- Waste of resources such as un-stain-removed clothes, laundry detergent and time.

Needs
Yehseul is looking for an easy and dependable way to figure out what kind of stain is on her clothes before doing the laundry. Instead of guessing, she wants straightforward advice on

how to clean each stain the right way. Her main goal is to get rid of stains easily, keep her clothes in good shape, and avoid unnecessary waste.

Usage scenario

Yehseul noticed stains on her clothes and took a photo and uploaded to ‘Stainaway’ to know what this stain is and the laundry method. After Stainaway analyzed its stain and showed the appropriate laundry method, she successfully removed the stain and reduced stress related to stains and laundry.

1.3 Service concept

StainAway is an AI-based stain classification and washing recommendation service designed to support people who are inexperienced with laundry. The service helps users identify the type of stain on their clothing and provides appropriate cleaning guidance before washing. Users can upload stain pictures on web or mobile applications. The system analyzes the image using an image classification model and categorizes the stains into ink, tomato sauce (ketchup), chocolate, coffee, wine, juice , blood and dirt. Furthermore, by showing accuracy(%), it is planned for users to understand results and trust. Although ‘clean’ is not shown as output, we added it in ML to divide stain and clothes’ pattern shadow and fabric texture. It is important to decrease errors and increase service reliability.

1.4 KPIs defined

KPI	Target	Description
Accuracy	≥ 85%	Correctly classifies the type of stain (coffee, oil, ink, etc.)
Latency	≤ 2.0 sec	Time taken from image to result output
Robustness	≥ 80%	Maintains stable accuracy under different lighting and fabric types
Recommendation quality	≥ 4.0 / 5.0	User rating on how helpful the cleaning method was
Usability	≥ 85% satisfied users	Percentage of users who found the service easy to use
Dataset expansion	+20% per iteration	Rate of dataset growth after each training cycle

2. Data

2.1 Collection Methodology

The goal of this project is to develop an AI model—**StainAway**—that classifies stains on fabric surfaces and recommends appropriate washing methods automatically.

To achieve this, a dataset of approximately **1200 images** was constructed across **9 classes**: **clean, coffee, wine, ink, blood, chocolate, tomato_sauce(ketchup), dirt/mud, juice**

Data collection was conducted using two primary approaches:

1) Directly Captured Images (≈200 original samples)

Images were captured with a smartphone camera under diverse conditions:

- Various lighting environments: indoor LED, fluorescent lights, natural daylight, shaded conditions, direct and indirect lighting
- At least **20 original images per class** were captured to reflect real-world variation

2) Online Image Acquisition (≈1000 samples)

Images were gathered from:

- Image search engines
- Google search (via SERPAPI)

Keyword-based queries such as *"coffee stain on fabric," "ink stain cloth," "mud stain textile"* were used. Non-fabric backgrounds were removed, and duplicates or low-resolution images were excluded.

2.2 Sources and Licenses

The dataset follows the licensing policies below:

- **Directly captured photos:** Free for academic project use with no copyright issues
- **Google / online images:**
    - SERPAPI allows research and non-commercial use
    - Images were used with original post references maintained, complying with fair-use principles

The project adheres strictly to research and educational use data requirements.

2.3 Labeling Process

Labeling was performed using Roboflow and a directory-based folder labeling system.
Steps followed:

1. All images were manually reviewed to remove mislabeled, irrelevant, or noisy samples
2. Each class was curated to include variations in lighting, texture, and stain shape
3. Because the task is image classification, bounding boxes were unnecessary
4. Folder names served as class identifiers, reducing labeling overhead and minimizing errors

2.4 Diversity Analysis

Model performance heavily depends on dataset diversity. We assessed diversity across several dimensions:

Lighting Diversity

- Indoor warm LED
- Fluorescent environments
- Natural sunlight / shadows

Lighting variations are critical because stain colors and textures change dramatically under different illumination.

Fabric Surface Diversity

- White cotton T-shirts
- Gray/black fabric backgrounds
- Smooth and rough materials

This allows the model to generalize across clothing and household textiles.

Stain Pattern Variability

Different stain classes exhibit consistent visual patterns:

Class	Characteristics
Coffee	Brown ring-shaped edges, diffused gradients
Blood	Dark reddish tone, irregular spread patterns
Dirt/Mud	Granular texture, speckled surface
Ketchup	Oily highlights, irregular shape
Ink	Very sharp boundaries, highly saturated regions
Wine	Uniform red tone, smooth bleed patterns
Chocolate	Dark viscous center with soft edges
Juice	Semi-transparent color, gradient edges

Augmentation

Offline augmentation was applied to compensate for limited real-world samples:

- Brightness / contrast changes
- Color jitter
- Zoom in/out
- Minor rotations (±5°)
- Horizontal flipping

These augmentations simulate realistic variations in smartphone photography and improve model robustness.

2.5 Data Sheet Summary

Item	Description

Total image count	~1200
Number of classes	9
Sources	Direct capture + web images (non-commercial)
Labeling method	Directory based classification
Augmentation	Color jitter, rotation, brightness adjustment
Purpose	Stain classification + washing guidance
Limitations	Strong light diversity, Similar color
Future improvements	Light adjustment, Larger datasets, Diverse color variations

3. Model Development Section

3.1 Tool / Framework Choice

Ultralytics **YOLOv8 Classification** (yolov8n-cls).
Reason

- Mobile-friendly and extremely lightweight → suitable for real-time inference on HuggingFace Spaces
  Optimized architecture for image classification
- Built-in extraction of softmax probabilities
- Easy to train in Google Colab
- PyTorch-based → flexible for customization
- Strong performance even with small datasets
- Self image augmentation

Deployment was implemented using Gradio + HuggingFace Spaces, enabling users to upload images and instantly receive stain type and washing instructions.

3.2 Training Process

Overall training pipeline:

1. Dataset split into train/val = 80/20
2. Image resizing: 224×224
3. Batch size: 32
4. Epochs: 30
5. Optimizer: SGD or Adam (auto-selected)
6. Loss: CrossEntropyLoss
7. Backbone: YOLOv8n-cls pretrained weights
8. Lighting normalization applied at inference time (normalize_lighting(

Entire training command:
3.3 Evaluation Metrics

Key evaluation metrics: Accuracy

Overall proportion of correctly predicted images: Top-5 Accuracy
Important because stain classification can be ambiguous; the correct label often appears within top-5 predictions. Fig 1 shows which classes are most frequently confused per-class precision & recall reveals how confidently the model predicts each stain type

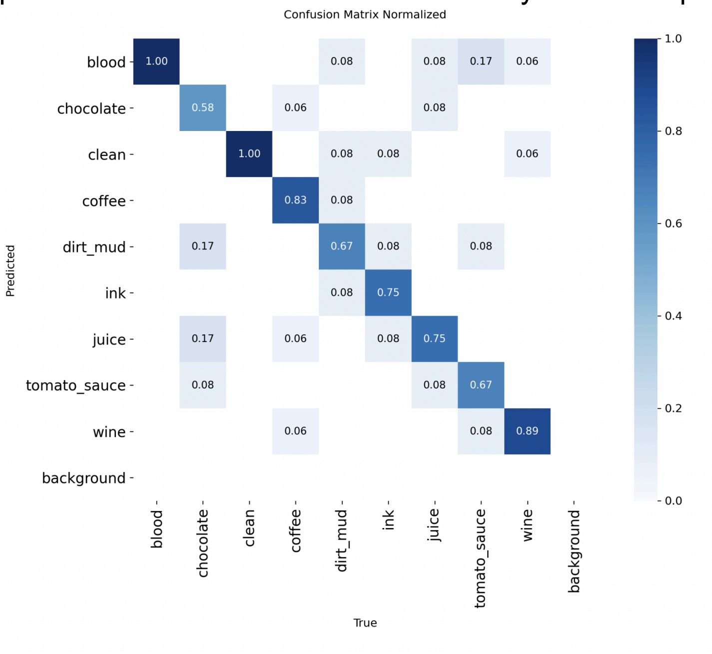


Fig 1(Confusion Matrix)

- Clean: 99.3% confidence
- Coffee: 99.6% confidence
- Wine: 99.9% confidence
- Ink: 98.7% confidence
- Tomato_sauce: 95.0% confidence
- Dirt_mud: 94.4% confidence
- Juice: 80.0% confidence
- Blood: 51.3% confidence
- Chocolate → Misidentified as juice (53.9% confidence)

3.4 Performance Results (Fig 2)

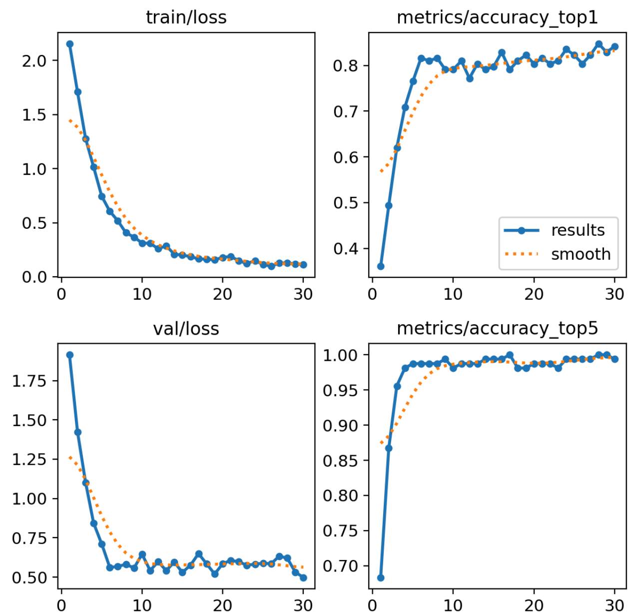


Fig 2

- Validation accuracy: 84.81%
- Top-5 accuracy: 100%

Strengths:

- Excellent performance for stains with strong color contrast
- High accuracy on clean backgrounds (white cotton)
- ink, coffee, blood show especially strong classification confidence

3.5 Error Analysis

Four major error type identified	Examples	
1. Lighting variation issues	• Tomato_sauce misclassified as chocolate in dim lighting • Yellow indoor lighting causing confusion between wine and blood	Solution: Strengthened the lighting normalization during inference
2. Influence of fabric background color	• Stains photographed on black fabric had blurred boundaries • Some stains became harder to distinguish due to low light absorption→ Future datasets should include more diverse backgrounds	
3. Color-similar classes	• coffee ↔ dirt/mud • tomato_sauce ↔ wine	Reason: Similar hues but different textures → challenging for models with limited texture cues
4. Underrepresented classes	Classes with fewer original photos (e.g., tomato_sauce, juice) relied heavily on augmentation → Performance was slightly lower on real-world test images.	

4. Service Implementation

4.1 Platform Choice rationale

WEB

HuggingFace

Reason

- **Easy Model Sharing and Deployment**
  Users can upload, version, and share models effortlessly, making collaboration and reproducibility much easier.

- **Integration with Popular Frameworks**
  Hugging Face integrates well with PyTorch, TensorFlow, JAX, and ONNX, providing flexibility across different ML workflows.
- **Rapid Prototyping and Experimentation**
  Pre-trained models and ready-to-use pipelines enable fast prototyping for research and real-world applications.
- **Open-Source Ecosystem**
  Hugging Face strongly supports open-source development, allowing researchers and developers to freely access, use, and contribute to models and libraries.

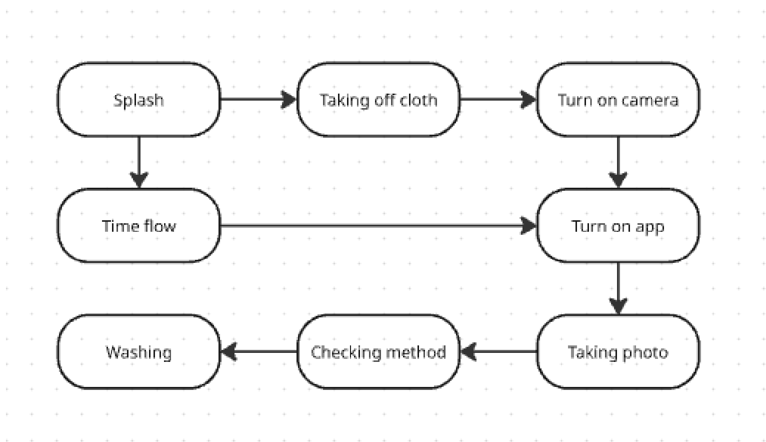
APP

Thunkable

Reasons

- WebView interworking was the most intuitive. It could load URLs directly with only one Web Viewer component.
- It is based on block coding, so fast repetition was possible.
- It had a structure where the user takes a picture or selects a picture in the gallery.
- It modifies the code and design and can be tested on the phone in real time.

4.2 UX Flow



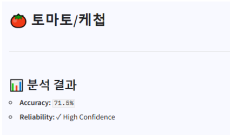
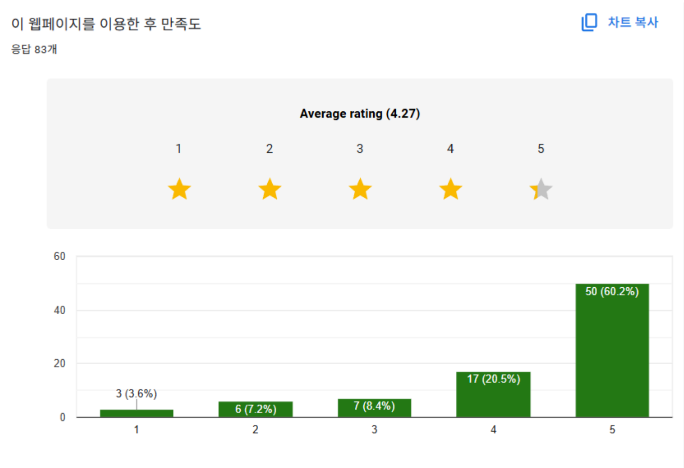
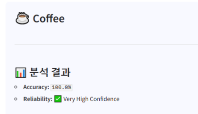
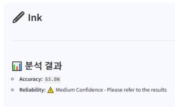
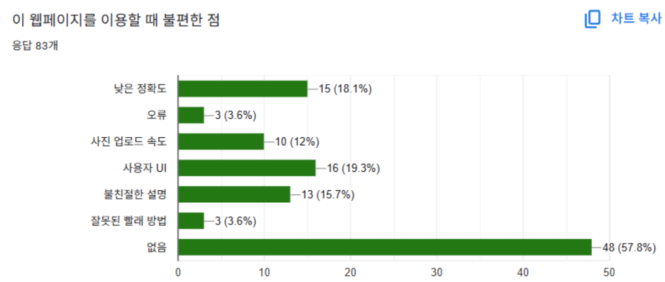
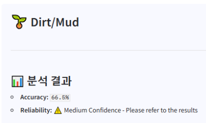
4.3 Deployment Process

Deployment was implemented using Gradio + HuggingFace Spaces, enabling users to upload images and instantly receive stain type and washing instructions. Users can also easily access the web page by creating a QR code. In the case of the web, they can use the generated URL in your browser or use the 'Add to Home Screen' function to use it like an app.

4.4 User testing results

Quantitative Data **Qualitative Data**

Photo	Result	Survey

<table>
<tr>
<td></td>
<td>
</td>
<td rowspan="2">
</td>
</tr>
<tr>
<td></td>
<td>
</td>
</tr>
<tr>
<td></td>
<td>
</td>
<td rowspan="2">
</td>
</tr>
<tr>
<td></td>
<td>
</td>
</tr>
</table>

5. Discussion

5.1 What worked well	**5.2 What didn't work**
YOLO-CLS v8 model based stain classification model shows high accuracy (Validation accuracy: 84.81%)	Even if it is the same stain, result is sometimes different depend on angle, lights, picture sizes
Autonomic data collection (via SERPAPI) + YOLO based Cropper make possible to collect learning data	Automated image collection and cropping can introduce mislabeled or low-quality samples, which can be harmful to model accuracy, especially in small datasets

Home screen, result page, laundry guide UI fully implemented (web/app)	The model hasn't been tested enough with real users, so its performance in real-world situations is still uncertain.
Camera input, upload function, inference pipeline connected; result rendering stable	With only 50-100 images per stain, the model struggles to reliably tell apart similar-looking stains.

5.3 Lessons learned

1. **Understanding what the model actually learns**
   YOLOv8n-cls does not rely on manually defined rules or simple visual statistics. Instead, it learns features automatically through convolutional layers. These features include patterns related to color differences, texture, and the shape of stains. Realizing this helped us understand that the model's decisions come from learned visual patterns, not from explicit instructions.
2. **Limits of a lightweight classification model**
   Since we used the nano version of YOLOv8, the model has limited capacity compared to larger variants. This means it cannot easily capture very subtle or complex stain differences. Through training and evaluation, we learned that model size and feature capacity directly influence classification accuracy.
3. **Dataset design and model perspective**
   In theory, understanding that YOLOv8-cls learns deep visual features should guide dataset collection toward diverse stain appearances, lighting conditions, and backgrounds. However, our group mainly collected copyright-free images from Google without a strict data strategy. This gap made us realize how important it is to align dataset design with what the model focuses on.
4. **Learning from accuracy limitations**
   Although we did not apply advanced dataset curation, observing low or unstable accuracy was still informative. Accuracy issues helped us identify where the model struggled and made the learning process more visible. This experience improved our understanding of how data quality and feature learning affect model performance.
5. **Key takeaway from the assignment**
   This assignment showed us that successful machine learning is not only about writing correct code. It also requires understanding the model's feature learning mechanism and preparing data accordingly. Even with a simple dataset, analyzing accuracy results helped us build a more realistic understanding of how image classification models behave.

5.4 Ethical considerations

1. **Informed Decision-Making:**
   Displaying the confidence level or accuracy percentage for each classification allows users to make better decisions about whether to trust the app's recommendation or seek manual/alternative advice, especially for low-confidence cases.
   Example: *"This stain is classified as Coffee with 92% confidence."*
2. **Building User Trust:**
   Transparency in model accuracy demonstrates professionalism and a commitment to user safety, showing that you acknowledge the possibility of error and care about user outcomes.
3. **Legal Protection (Indemnification):**

Since misclassification or incorrect laundry advice can cause permanent clothing damage, an explicit indemnification clause in the user agreement is essential. This clause should state that while best efforts are made, the service is not legally responsible for any losses or damages due to the recommendations.

4. **Sample Indemnification Clause:**
“While this app/website uses advanced AI to recommend stain types and cleaning methods, results are based on image analysis and provided for informational purposes only. We do not guarantee complete accuracy. Users accept full responsibility for their chosen actions, and the service and its creators cannot be held liable for any direct or indirect damages resulting from the use of these recommendations.”

5.5 limitations

Category	**Limitation description**
User testing scale	Since testing users was only 3, it didn’t handle various users’ (age, clothing styles, using devices) opinion or perspectives.
Multi stain handling	The system is not yet designed to handle multiple or overlapping stains in a single image, which occurs frequently in real-world laundry scenarios.
Device & camera variance	Differences in smartphone cameras, resolutions have not been accounted for in model training, which could impact consistency across devices.
Dataset diversity	The current dataset is biased toward clean, well-lit images. It lacks variety in lighting, fabric types, stain spread patterns, and partial stains commonly found in real life.

4.6 Future works

Technical advancement

- The performance of the current model is biased against stains shot in a general lighting environment. Since there are various variables such as night, various color lighting, backlight, and blur in the actual user environment, it is necessary to strengthen the generalization performance of the model by securing additional images taken in this situation.
- Current models have high accuracy but somewhat long loading speeds. To improve this, we apply lightweight model structures such as TensorFlow Lite conversion and MobileNet and EfficientNet, and reduce the amount of computation by using quantization techniques. Through this, the user's convenience is reliably improved.
- If you choose a camera, there may be confusion because it is not a general design. It reduces confusion for users by choosing a popular method that is easy for users to understand.

Adding functions

- Currently, only the washing method is printed, so user persuasion is insufficient. In order to present the recommended basis together, the detected stain color and feature are specified. Based on this, a simple extendable output for the

recommended washing method is added. This clearly communicates why this method should be used, thereby increasing service reliability.
- Since the current UI style looks slightly different for each screen, a unified design guideline is established. Text size, color, and alignment criteria are organized, and all screens are modified to maintain a consistent tone and manner. It also adds visual recommendation intensity display and key sentence emphasis to visually understand the results at a glance.

6. Conclusion

This project presented *StainAway*, an AI-based stain classification and washing recommendation service designed to support users with limited laundry experience. By combining image classification with practical cleaning guidance, the system aimed to reduce guesswork in stain removal and prevent unnecessary fabric damage. Throughout the project, we focused on building an end-to-end pipeline, from data collection and model training to deployment on both web and mobile platforms.

One of the key outcomes of this work was demonstrating that a lightweight image classification model can still achieve meaningful performance in a real-world service context. Using YOLOv8n-cls, the model reached a validation accuracy of 84.81% and a Top-5 accuracy of 100%, showing strong performance for stains with clear color and texture patterns such as coffee, ink, and wine. In addition, the integration of confidence scores allowed the system to communicate uncertainty to users, supporting more informed decision-making rather than presenting results as absolute truths.

Beyond model performance, this project emphasized the importance of system-level thinking in applied machine learning. Building the service required aligning dataset design, model capacity, user interface, and ethical considerations into a coherent whole. Through this process, we learned that even small choices, such as lighting conditions in data collection or UI feedback design, can significantly influence both model behavior and user trust. The experience highlighted that machine learning systems should be evaluated not only by accuracy metrics, but also by how transparently and responsibly they interact with users.

While the current version of StainAway has limitations, including a small user testing scale and reduced performance for visually similar stains, these constraints provided valuable insights into future improvement directions. Overall, this project reinforced the idea that effective AI services are built through iterative refinement, careful data understanding, and thoughtful integration between technical performance and user-centered design. StainAway serves as a practical foundation for further development toward a more robust and reliable everyday AI assistance tool.

7. Appendices

Appendix A. Full error board

Error Case ID	Predicted Label	True Label	Cause	Notes
CH-01	Juice	Chocolate	Chocolate was widely spread and thinned	Lost dark center, appeared diluted
CH-02	Ketchup (Tomato sauce)	Chocolate	Color shifted to reddish-brown under warm lighting	Yellow indoor light affected hue
CH-03	Coffee	Chocolate	Chocolate absorbed into fabric fibers	Flat brown appearance
CH-04	Wine	Chocolate	Low-light conditions reduced texture visibility	Dark uniform tone
CH-05	Dirt/Mud	Chocolate	Thick or dried chocolate formed rough texture	Texture similar to soil
CH-06	Dirt/Mud	Chocolate	Low contrast on dark fabric background	Boundary unclear
WI-01	Juice	Wine	Indoor lighting altered red tone	Appeared lighter than typical wine
WI-02	Coffee	Wine	Dim lighting darkened stain color	Texture details lost

JU-01	Wine	Juice	Juice highly absorbed into fabric	Color became too light
CO-01	Dirt/Mud	Coffee	Similar brown hue and irregular spread	Texture overlap
DI-01	Coffee	Dirt/Mud	Mud dried smoothly on fabric	Looked like dried coffee stain

Appendix B. Code Snippets

This appendix provides representative code snippets used for training and inference in the StainAway system. The snippets illustrate the overall structure of the model pipeline rather than full implementation details.

B.1 Model Training (YOLOv8 Classification)

```
from ultralytics import YOLO

# Load pretrained YOLOv8 classification model
model = YOLO("yolov8n-cls.pt")

# Train the model
model.train(
    data="stain_dataset",
    epochs=30,
    imgsz=224,
    batch=32,
    device="cuda"
)
```

The model was trained using a directory-based dataset structure, where each folder name represented a stain class. Pretrained weights were used to improve convergence on a relatively small dataset.

B.2 Inference and Prediction

```
# Run inference on a test image
results = model("input_image.jpg")

# Extract top predictions and confidence scores
top1_label = results[0].names[results[0].probs.top1]
top1_confidence = results[0].probs.top1conf
top5_predictions = results[0].probs.top5
```

The inference output includes the predicted stain type along with confidence scores. These values are later displayed to users to support informed decision-making.

B.3 Web Deployment (Gradio Interface)

```
import gradio as gr

def predict(image):
    results = model(image)
    label = results[0].names[results[0].probs.top1]
    confidence = results[0].probs.top1conf
    return label, float(confidence)

gr.Interface(
    fn=predict,
    inputs="image",
    outputs=["text", "number"],
    title="StainAway: Stain Classification Demo"
).launch()
```

This interface was deployed on HuggingFace Spaces, allowing users to upload images and instantly receive classification results.

Appendix C. User Test Scripts

This appendix outlines the user testing procedure and question script used to evaluate usability and clarity of the StainAway service.

C.1 User Test Scenario

1. The participant uploads a photo of a stain on their clothing.
2. The system analyzes the image and displays the predicted stain type and confidence score.
3. The participant reviews the recommended washing method.
4. The participant provides feedback based on their experience.

C.2 User Test Questions

1. What type of stain did you expect before using the system?
2. Did the predicted stain type match your expectation?
3. How clear was the confidence score shown on the result screen?
4. Was the washing recommendation easy to understand?
5. Would you trust this system for future laundry decisions? Why or why not?

C.3 Evaluation Focus

- Perceived accuracy of stain classification
- Clarity of result presentation
- Usefulness of washing recommendations
  Overall ease of use