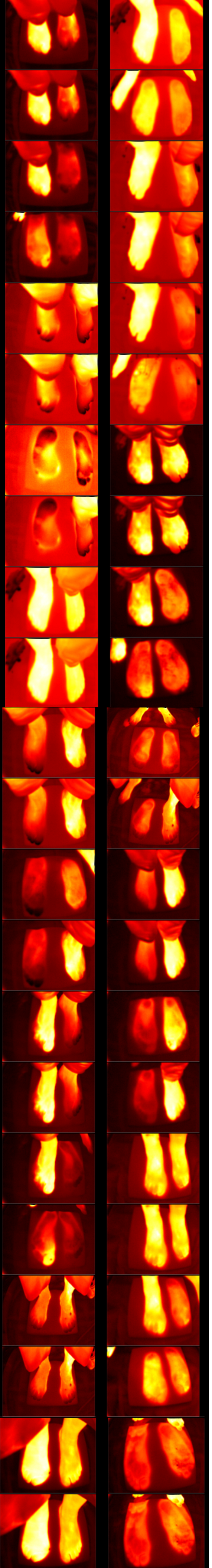


Design Report FUERA: Foot Ulcer Early Recognition Algorithm

Group 13: Kurt Salapare, Narendra Setty, Alex Duncan, Dan Gladkov, Hoang Pham

Table of Contents

1. Introduction

Diabetes is a widespread and growing disease that affects over 100 million people worldwide. It has many adverse effects on patients' health, one of which is diabetic foot ulcer (DFU). In particular, 34% of patients with diabetes develop DFUs, typically due to neuropathy and repetitive stress. Often leading to prolonged immobility, which requires extensive care and, in 25% of cases, results in some form of amputation. Patients who experienced DFUs have a 40% chance of recurrence within one year. These effects are grave but can be mitigated through early detection of DFUs and subsequent measures taken, such as offloading.

2. Problem Statement

The client for this project is ZGT Hengelo, a hospital in the Netherlands that is interested in developing an Early Diabetic Foot Ulcer Detection system. The client's Bath Mat system (explained later in the literature review) would use a thermal camera to record patients' feet temperatures daily with the goal of identifying ulcers in their early stages, thus, allowing for adequate measures to be taken. However, besides the physical device, a proof-of-concept for an algorithm is required that can algorithmically detect such wounds in early stages before complications occur, hence the need for the project.

One of the requirements of this project was to generate synthetic thermal images of feet in the pre-ulcer/developing ulcer state. This data is difficult to acquire due to the need for consistent measurements over a long period of time, and thus, such data does not exist. However, a dataset containing 30 images of patients with existing DFUs and 150 images of 15 healthy patients over 10 days - both Plantar (bottom of the foot) and Dorsal (top of the foot) - were given as a baseline to be used to generate the synthetic images.

Moreover, the second requirement of the project relates to creating an algorithm to detect DFUs in their developing stage. The synthetic data from the first step can thus be used to test the algorithm sufficiently. Afterwards, it can be used to detect developing ulcers, allowing patients to take the appropriate measures, preventing the further development of DFUs, which in turn helps reduce the patient load of hospitals and reduces the patients' costs related to healthcare.

In the end, the functionality of the generation of synthetic thermal images of feet in a developing ulcer state was successfully implemented. This was achieved by taking a healthy patient's foot, generating a wound heat mask that increases over a certain period of days.

Furthermore, an algorithm - named FUERA (Foot Ulcer Early Recognition Algorithm) - was developed, which is consistently able to detect DFUs in their early stages. This is done by measuring temperature differences both in opposite regions of both feet and in regions on individual feet. This algorithm can then be used for a proof of concept product, which can be placed in patients homes, used for early ulcer detection while also collecting real data of patients' ulcer development.

3. Requirements Specification

Prior to beginning with the design and implementation of our project, a comprehensive list of requirements to guide the development of the system was compiled. Given the nature of the project, and its focus on backend processing rather than a user-facing web-interface or UI, emphasis was placed solely on functional and technical requirements. Non-functional requirements such as usability, response-time or accessibility were deemed outside the scope.

3.1. Core Functional Requirements for Minimum Viable Product (MVP)

Initially, the following essential capabilities to deliver a functional MVP aligned with client expectations were defined:

- The system should be capable of accepting thermal foot scan data in MATLAB file format.
- The system should simulate progressive wound evolution over a period of days, increasing in size from Stage 0 (no visible ulcer) to Stage 1 (early formation).
- The system should be able to analyze sequential thermal thermal images to identify patients at risk of developing foot ulcers, flagging cases based on a ≥2.2 degree difference between contralateral regions of feet (anatomically same regions on both).

3.2. Enhanced Requirements

As the aspects of wound generation and early detection were delved into, a list of requirements following interviews with the client and further research were compiled. These requirements supplemented the ones previously collected and included the following:

- The system should be able to produce wounds of diverse shapes and randomly place them across varied regions of the foot.
- The system should be able to support wound generation and classification on various foot types and dimensions.

3.3. Supplementary Early Detection Methodology

Following successful implementation of the primary requirements, the client proposed an alternative detection paradigm for integration:

- The system should be able to perform an early detection of an ulcer by training the algorithm on images of individual feet to form a baseline and analyzing regional temperature differences over sequential days.

3.4. Visualization and Explainability Requirement

Finally, to allow the client and possible future patients to view the processing steps and reasoning used by the algorithm, the following requirement was also created:

- The system should be able produce annotated images showing progression of synthetic wound development as well as regions and rationale used for early ulcer detection.

4. Related Work

Once the requirements were collected, a literature review of state-of-the-art papers was conducted to understand the current and most used methods in wound generation and early diabetic ulcer detection. This analysis specifically aimed to uncover limitations and areas for enhancement to inform the subsequent design and implementation phases.

4.1. Diabetic Foot Wound Locations

Prior to investigating current methodologies used for image generation which could be used to generate wounds, research was also done to collect information regarding the most likely areas of the foot for wound formation. The paper by Cowley *et al.* (2008), provided the necessary information which involved examining 3040 feet and showing each region of the foot with the amount of ulcers that were found in those regions. It was observed that regions such as the hallux, toes, heel had a much higher rate of ulcer development when compared to the midfoot region. Figure 1 shows the different regions of the foot with their rate of ulcer formations which informed the design choices for wound generation locations. The diagram of the foot with the differently colored regions was made to add clarity to the theory.

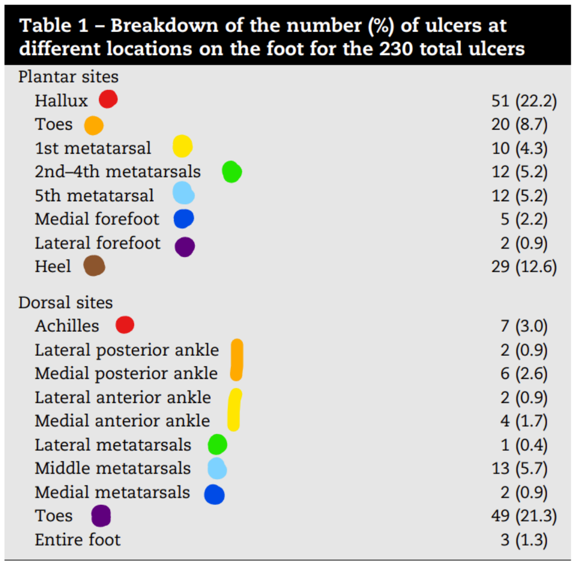


Table 1 – Breakdown of the number (%) of ulcers at different locations on the foot for the 230 total ulcers

Plantar sites		
Hallux	🔴	51 (22.2)
Toes	🟠	20 (8.7)
1st metatarsal	🟡	10 (4.3)
2nd–4th metatarsals	🟢	12 (5.2)
5th metatarsal	🔵	12 (5.2)
Medial forefoot	🔵	5 (2.2)
Lateral forefoot	🟣	2 (0.9)
Heel	🟤	29 (12.6)
Dorsal sites		
Achilles	🔴	7 (3.0)
Lateral posterior ankle	🟧	2 (0.9)
Medial posterior ankle	🟧	6 (2.6)
Lateral anterior ankle	🟨	2 (0.9)
Medial anterior ankle	🟨	4 (1.7)
Lateral metatarsals	🟢	1 (0.4)
Middle metatarsals	🔵	13 (5.7)
Medial metatarsals	🔵	2 (0.9)
Toes	🟣	49 (21.3)
Entire foot		3 (1.3)

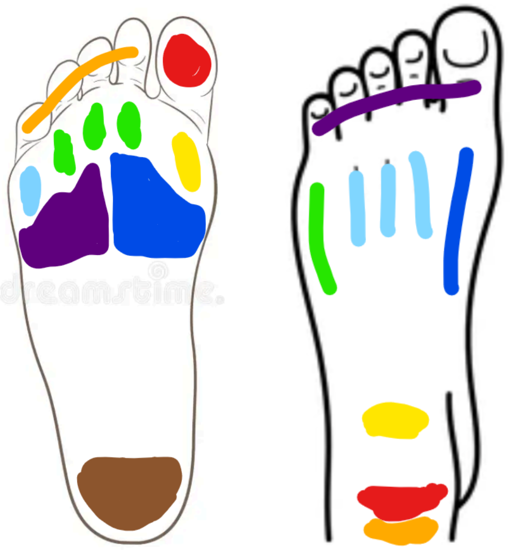


Figure 1: Probability breakdown of common wound spots by Cowley et al. (left) with corresponding color-coded locations (right)

4.2. Potential Image Generation Strategies and Their Limitations

A potential strategy for synthesizing augmented thermal image data was by utilizing a StyleGan's Model and incorporating it with transfer learning and vector interpolation. This approach was going to be the most mathematically sophisticated one for generating thermal images simulating ulcer progression. The main idea was to capitalize on StyleGAN's core strength: its high proficiency in learning a highly organized, disentangled latent space. Essentially, the main idea behind this methodology would be to do the following: after training a model on both healthy and unhealthy (ulcer present) feet, specific dimensions within the latent space would directly correspond to high-level features such as the stage in ulcer progression and temperature patterns, independent of features like foot shape or foot size. This capitalizes on the following idea: calculating / finding the "Ulcer Vector" by subtracting the latent vector of a foot that is healthy from an ulcerated foot. This resultant vector will mathematically encode the precise visual and thermal changes necessary for ulcer generation. In order to simulate progression, we would start with the latent vector of a healthy foot and then progressively increment by adding small fractions of the Ulcer Vector. Feeding these latent vectors back into the trained StyleGAN generator would result in a very realistic progression of ulcer development.

However, during the early research of this approach, a major roadblock completely stopped this approach from being possible - the dataset provided for this project is too small. For this approach it is recommended to have 50,000 to 100,000 images and at the bare minimum 10,000 to 20,000 for training, however there were only 150 healthy and 30 unhealthy images provided which would lead to discriminator overfitting (Karras et al., 2020). This resulted in completely abandoning this approach and shifting towards a more algorithmic approach.

4.3. Early Detection

Research into early detection systems for diabetic foot ulcers has yielded multiple methodological approaches. Brindha *et al.* (2024) developed a portable prototype scanner integrating a high-resolution thermal camera for DFU detection targeting the plantar region (flowchart seen on Figure 1). Thermal images were captured as rainbow-colored thermograms after 15-20 minutes of patients acclimation to normalize blood flow. A lightweight MobileNet CNN was trained on pixel values from a dataset of plantar images of the Plantar Image Database consisting of 122 diabetic patients and 45 healthy patients, following the standard Deep-Learning pipeline: splitting, training/validation/testing, and color-mapped post-processing for real-time ulcer classification/severity.

Another methodology was implemented by Alshayeji *et al.* (2023) using a bag-of-features technique. It began with pre-processing data from the Plantar Image Database, where raw images were normalized, contrast enhanced and converted to thermal images. Following this local feature descriptors were extracted and clustered using a bag-of-features technique using k-means producing fixed-size vectors for each image. These features were used to train and test several machine learning classifiers and validated through tenfold cross-validation for which the SURF-BOF-based-SVM achieved the best accuracy of 97.81%.

4.4. Limitations with State-of-art Early Detection Methods

While numerous studies have aimed to develop pre-ulcer detection systems, a common methodological shortcoming undermines their potential. The reliance on datasets of single, point-in-time images from distinct patients ignores critical temporal factors. Consequently, these models cannot learn to distinguish pathological signs from normal daily temperature variations and transient anomalies, which is essential for accurate, long-term monitoring. Furthermore, approaches such as that of Alshayeji *et al.* (2023), although reporting high accuracy, were developed using a very limited dataset which is an approach that is generally unsuitable for robust model training. Even with tenfold cross-validation, such models are highly susceptible to overfitting and may fail to generalize effectively to unseen data. Deep learning techniques were also deemed impractical for this project, as they typically require thousands of images to achieve reliable performance, whereas the provided dataset was considerably smaller.

4.5. Paper Provided by the Client

In addition to the research into different studies, the client provided an explorative pilot study conducted by Zoetlief (2023). It took place on the Bath Mat, a novel thermographic device designed for domestic early detection of diabetic foot ulcers by capturing full-foot temperature distributions (see Figure 2). The dorsal side of the foot was imaged directly, while the plantar side was assessed indirectly, via thermal footprints left on the map after stepping off, addressing limitations of previous studies that focused solely on the plantar side with a few measurement points. In a study conducted with 20 healthy patients and 30 patients with DFUs, thermal footprints showed high similarity to direct plantar images including hotspots and coldspots. The study provided this project with the dataset of thermal images consisting of 15 healthy patients over a period of 10 days and 30 images of different DFU patients, collected using the same client's Bath Mat thermography device. Additionally, this study provided confirmation regarding the theory about 2.2°C temperature difference on contralateral regions of the foot as a warning threshold for a possible ulcer formation.

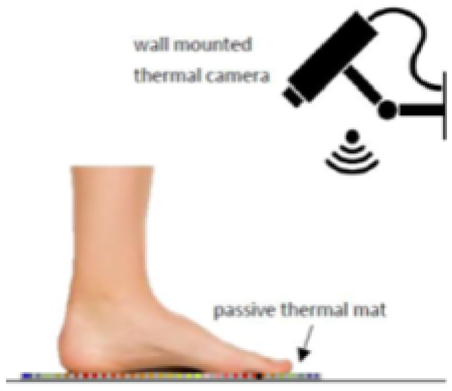


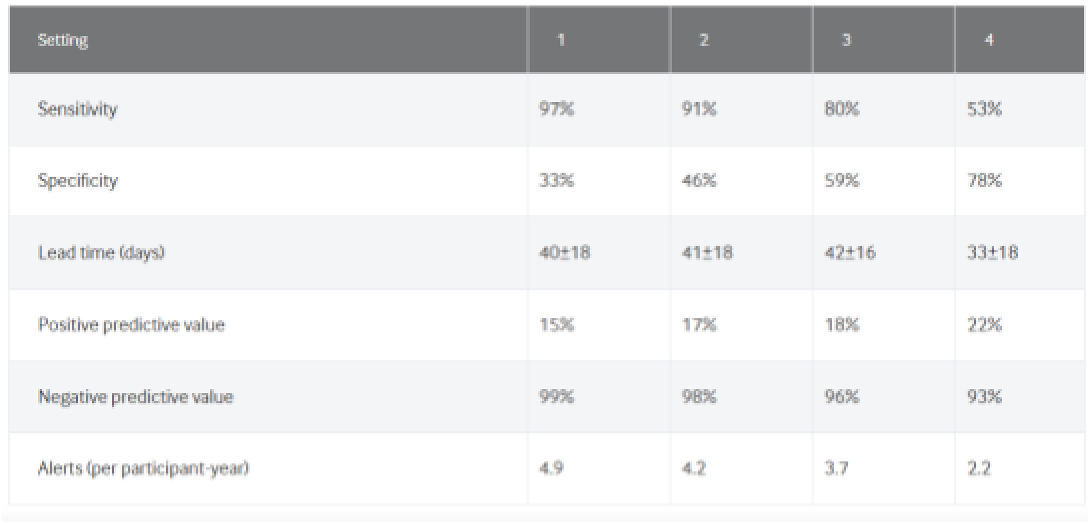
Figure 2: Schematic of the Bath Mat device

4.6. Researcher Interview

Another interesting and novel approach for early detection of ulcers was implemented by Lavery *et al.* (2019). This study demonstrated the use of a single foot in order to detect these wounds which wasn't seen in other studies. It proposed dividing the foot into different regions and then comparing the maximum temperature and the minimum temperature among the six regions which is termed "ipsilateral temperature range" (ITR). However, details regarding specific threshold values were not included in the study, which resulted in a meeting with one of the lead authors, Dr. Lawrence Lavery. The key takeaways from the meeting were as follows:

- Specific threshold values could not be shared as the paper was the intellectual property of the company that funded the research (see Figure 3).
- The days leading to ulcer formation was altered to 30-40 after initially being set to 10-15.
- The only source of information backed by reliable sources as an early warning mechanism for ulcers is the 2.2 degree temperature difference between contralateral regions of the foot.

Despite not getting the specific threshold values from Dr. Lavery, the afore-mentioned valuable insights were gained which were incorporated into the final design.



Setting	1	2	3	4
Sensitivity	97%	91%	80%	53%
Specificity	33%	46%	59%	78%
Lead time (days)	40±18	41±18	42±16	33±18
Positive predictive value	15%	17%	18%	22%
Negative predictive value	99%	98%	96%	93%
Alerts (per participant-year)	4.9	4.2	3.7	2.2

Figure 3: Results from paper by Lavery et al (specific details of settings couldn't be provided for confidentiality reasons)

5. System Pipeline Overview

After having concluded collecting and forming the requirements, a pipeline was constructed for the proposed system to guide the implementation of the product in several steps. This section presents the high-level overview of the devised pipeline, both in visual as well as textual format.

First and foremost, the previously unfamiliar MATLAB files of foot thermography had to be analyzed, studied and unpacked to understand their structure. Moreover, due to the nature of the files provided by the client, the data had to be made presentable. This initial step of the project focused on setting up the logic of temperature values plotting of the MATLAB files as an image, the method that was then used throughout the pipeline to display images for demonstration and testing purposes.

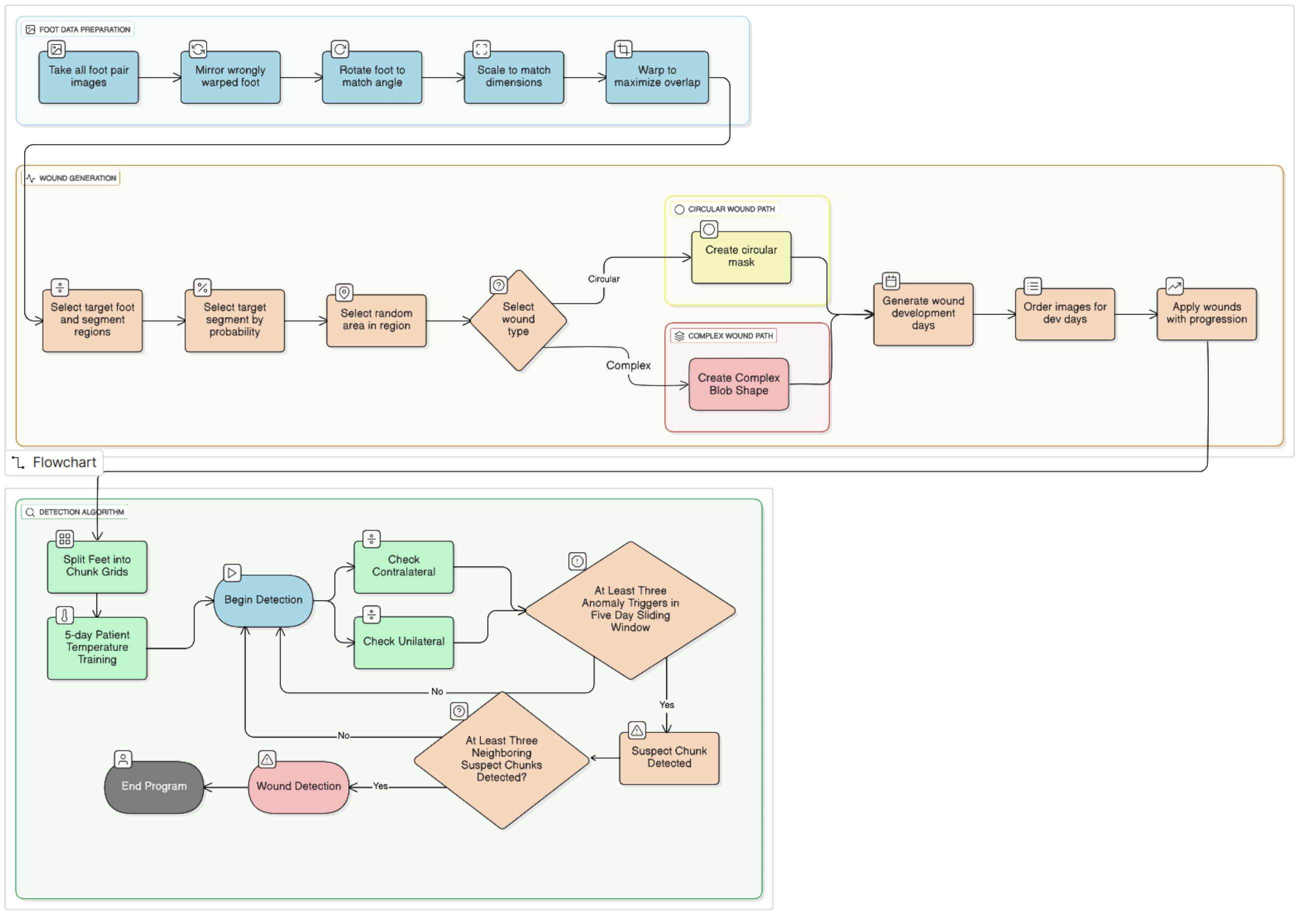

FOOT DATA PREPARATION
Take all foot pair images
Mirror wrongly warped foot
Rotate foot to match angle
Scale to match dimensions
Warp to maximize overlap
WOUND GENERATION
Select target foot and segment regions
Select target segment by probability
Select random area in region
Select wound type
Circular
Complex
CIRCULAR WOUND PATH
Create circular mask
COMPLEX WOUND PATH
Create Complex Blob Shape
Generate wound development days
Order images for dev days
Apply wounds with progression
Flowchart
DETECTION ALGORITHM
Split Feet into Chunk Grids
5-day Patient Temperature Training
Begin Detection
Check Contralateral
Check Unilateral
At Least Three Anomaly Triggers in Five Day Sliding Window
No
Yes
Suspect Chunk Detected
At Least Three Neighboring Suspect Chunks Detected?
No
Yes
Wound Detection
End Program

Figure 4: FUERA pipeline overview

Subsequently, the devised implementation and the background of the project provided ground for the complete FUERA pipeline to be split into two major consecutive areas (see Figure 1 for a visual depiction):

1. Synthetic image generation. The provided 10-day thermography of 15 healthy feet was taken as the baseline for synthetic images. Each set of feet was first put through "foot correction" - scaling, rotating and warping one foot to match the other - to eliminate the observed issues with the camera perspective of given thermal files. Then, both feet were segmented into three regions: upper foot, mid foot and heel.

These initial steps primed the baseline for the application of a synthetic wound (diabetic foot ulcer), which was initialized randomly on either foot and probabilistically at one of the three regions. To facilitate natural wound growth, each of the 15 sets of 10-day healthy scans were extended via sequential image repetition to (at most) 40-day image sets, covering the phases of initial wound absence, gradual wound growth via predefined wound development variables, and later static wound presence.

2. Wound detection. An algorithmic approach was integrated with the outputs of the first step to run through each generated thermography set. FUERA utilizes the grid methodology as its foundation, by first splitting each of the feet into a grid of temperature chunks, and then analyzing each one day by day.

FUERA comprises two methods of anomaly checks. The primary analysis compares the contralateral chunks on both feet for any temperature irregularities, while the secondary one monitors the mean temperature of chunks only on individual feet. The crucial functionality that makes this algorithm possible and less prone to false positives is the incorporated sliding window of 5 days, only in which three neighboring chunks have to trigger for FUERA to actually conclude that a possible wound has been detected.

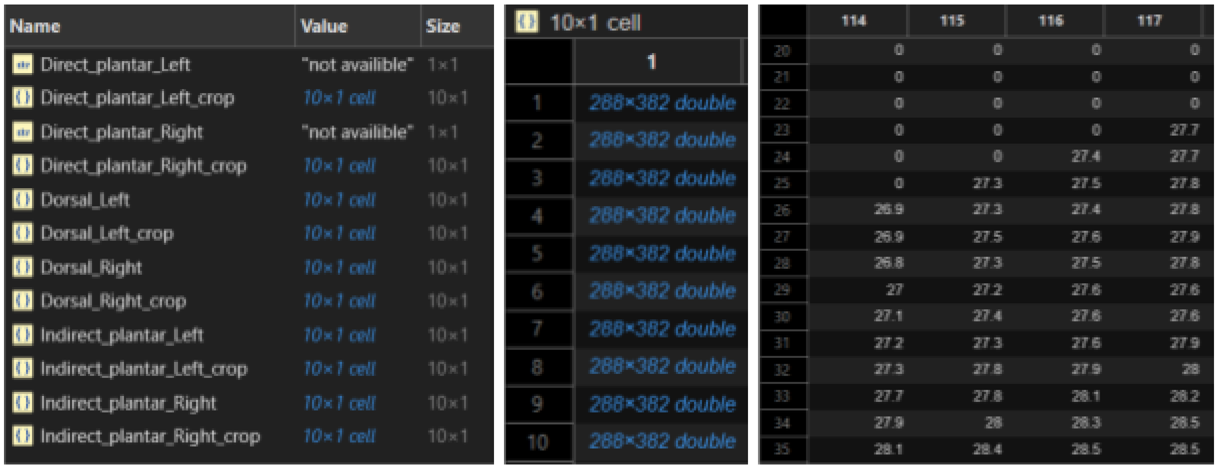
6. Detailed Design Overview

Upon the creation of the pipeline required by the system, the actual implementation of the product began. This section dives deeper into the technicalities of FUERA, and what specific functionalities and implementations were considered for the pipeline.

6.1. Data Familiarization + Visualization

As the preliminary step of understanding on the approach to the project, the provided files had to be studied, given that they are `.mat` files produced in the MATLAB environment, a previously unfamiliar domain. As previously mentioned, the files provided consist of both patients confirmed to have a DFU (30 files titled `pntN.mat`) and healthy individuals (15 files titled `gzN.mat`). As the scope of the project narrowed down to generating synthetic images by using healthy feet (more information in its respective Section 6.3), FUERA only utilizes the `gzN.mat` files. Furthermore, as shown on Figure 5, each of these files contains several values of the various versions of the taken thermal photographs.

For the algorithm, it was decided to utilize and process only the following values: `Indirect_plantar_Right_crop`, `Indirect_plantar_Left_crop`, `Dorsal_Right_crop` and `Dorsal_Left_crop`. The choice of indirect rather than direct for the plantar images stemmed



Name	Value	Size
Direct_plantar_Left	"not availible"	1×1
Direct_plantar_Left_crop	10×1 cell	10×1
Direct_plantar_Right	"not availible"	1×1
Direct_plantar_Right_crop	10×1 cell	10×1
Dorsal_Left	10×1 cell	10×1
Dorsal_Left_crop	10×1 cell	10×1
Dorsal_Right	10×1 cell	10×1
Dorsal_Right_crop	10×1 cell	10×1
Indirect_plantar_Left	10×1 cell	10×1
Indirect_plantar_Left_crop	10×1 cell	10×1
Indirect_plantar_Right	10×1 cell	10×1
Indirect_plantar_Right_crop	10×1 cell	10×1

10×1 cell

	1
1	288×382 double
2	288×382 double
3	288×382 double
4	288×382 double
5	288×382 double
6	288×382 double
7	288×382 double
8	288×382 double
9	288×382 double
10	288×382 double

	114	115	116	117
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	27.7
24	0	0	27.4	27.7
25	0	27.3	27.5	27.8
26	26.9	27.3	27.4	27.8
27	26.9	27.5	27.6	27.9
28	26.8	27.3	27.5	27.8
29	27	27.2	27.6	27.6
30	27.1	27.4	27.6	27.6
31	27.2	27.3	27.6	27.9
32	27.3	27.8	27.9	28
33	27.7	27.8	28.1	28.2
34	27.9	28	28.3	28.5
35	28.1	28.4	28.5	28.5

Figure 5: File structure of one `gzN.mat` (left), one inner value of one `gzN.mat` (middle), and one `double` element of one "`_crop`" inner values (right)

from the fact that an indirect photo is the thermography captured on the surface of the mat the patient was standing on, in comparison to the direct one being a photo of the patient's feet themselves. The former one has a higher accuracy of its temperature files due to its fixed and stable setup. Additionally, the cropped versions were used for the entire pipeline, as this alleviates the need to worry about the background of the images.

All of these values are a 10×1 cell, with one element corresponding to one day of photography. Each element is essentially the temperature array - the fundamental element FUERA is built up on - comprising a 288×382 table of temperature floats in degrees Celsius (°C) with surrounding zero values (cropped background in the "`_crop`" values).

The code snippet below and its output shown in Figure 6 were utilized as a debugging slot-in which helped in visualizing the available foot temperature arrays throughout the areas of the pipeline by plotting all of the array values as a graph with a temperature mapping.

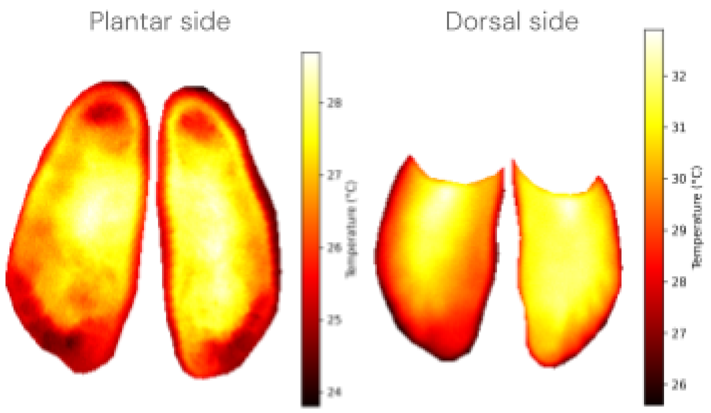


Figure 6: Baseline temperature array visualization

```python
fig, ax = plt.subplots(figsize=(8, 6))
    im = ax.imshow(temperature_array, cmap="hot", interpolation='nearest')
    ax.axis("off")
    ax.set_title(f"Day {day_index+1}: Foot for {pnt} Left Side")
    cbar = fig.colorbar(im, ax=ax, fraction=0.035, pad=0.04)
    cbar.set_label("Temperature (°C)")

    plt.show()
```

6.2. Foot Correction

6.2.1. Mirroring, Scaling & Rotation

The first geometric transformation that is applied to the thermal image of one foot (right foot) is mirroring. Since the feet are mirrored versions of one another, by horizontally mirroring the right foot, they can be compared. This was done using NumPy's `fliplr()` function that is inside the `mirror_horiz()` method. This is to ensure that locations on both feet are comparable.

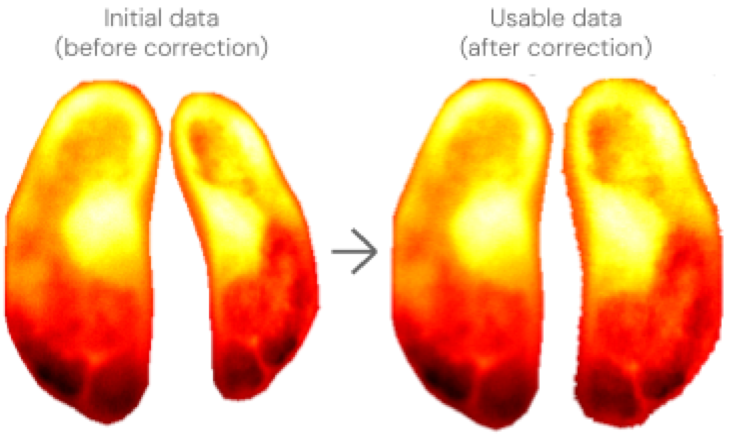


Figure 7: Foot correction applied on one set of feet

The next step after mirroring is rotation. This guarantees that the orientation of both feet matches as closely as possible. The algorithm tries different rotation angles between `-30°` and `+30°` and then calculates the overlap between the two feet and finds the optimal angle with the highest overlap, done in increments of `0.5°`. This helps to correct any small differences in the patient's stance or camera warping and prepares the image for further augmentations.

The final step is scaling. In order to have accurate pixel-wise comparisons, both feet were made the same size, by scaling the image vertically and horizontally. First, both images are put on a canvas the size of the biggest foot and then it computes new coordinates to resample the smaller foot using the method of nearest-neighbor interpolation. This is to make sure that the scaled foot's resolution and field of view match the other foot, while preserving the temperature values.

6.2.2. Warping

Warping is used to fine-tune the alignment of one foot with the other by applying small, smooth shape adjustments. The goal is to make both feet match as closely as possible. This is achieved by optimizing a displacement field (a map showing how each part of the image should move) using gradient descent to maximize the Intersection over Union (IoU) between their binary masks. In simple terms, the algorithm gradually adjusts one foot's shape until it overlaps the other as much as possible.

This process is implemented in `PyTorch`. Both binary foot masks are first converted into tensors, and a normalized coordinate grid (ranging from -1 to 1) is created to describe the image space. A small `3×3` grid of adjustable control points is then defined to guide how the image can stretch or shift. This grid is expanded to the full image size, forming a detailed map that determines how each pixel can move.

During optimization, the displacement field deforms one mask so it matches the other. The algorithm updates the grid in small steps to maximize IoU. To avoid unrealistic warping, the loss function also includes regularization terms that discourage large distortions and encourage smooth transitions across neighboring regions. The Adam optimizer runs for a fixed number of iterations, gradually refining the deformation grid until the two masks align closely.

After optimization, the best displacement field is saved as the final warping grid. This grid is then applied not only to the foot masks but also to the actual thermal images, ensuring they align spatially while preserving temperature information. To maintain temperature accuracy, each pixel's new value is taken from its nearest neighbor, similar to a k-nearest neighbor (kNN) interpolation.

The result of the 2-step foot correction preprocessing is a pair of spatially aligned thermal images with accurate temperature values, enabling precise comparison between corresponding regions of both feet (see Figure 7).

6.3. Synthetic Wound Generation

6.3.1. Foot Segmentation

Foot regions. Each foot image is split into three anatomically significant regions - the *upper foot*, the *mid foot* and the *heel* - by using the `segment_foot()` function:

```
def segment_foot(foot_arr):
    top, bottom = horizontal_split_by_percentage(foot_arr,
0.65)
    heel, mid_foot = horizontal_split_by_percentage(top,
0.4)
    return heel, mid_foot, bottom
```

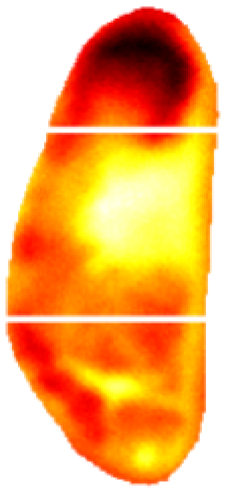


Figure 8: Two-split foot segmentation

Figure 8 demonstrates the three distinct regions of the foot. In essence, the segmentation performs a two-split step. The lower divider in the figure is the first split, which returns `top` 65% and `bottom` 35% of the foot. The upper divider is the second split, which is applied on the `top` segment of the previous split, returning the heel region (40% of the `top` segment) and the mid foot region (remaining 60%). The `bottom` segment is subsequently treated as the upper foot region. These calculations approximately provide the three considered regions in accordance with the literature review, with the majority of the upper foot region encompassing the toes, where most of the DFUs usually occur, and the heel region anatomically capturing the second-most frequent area under threat of ulcerations.

6.3.2. Wound Location

Following segmentation, the program picks two location parameters for each wound, first option being which foot of the two, and the second one being which region on that foot:

```
"apply_to_foot": rng_.choice(["left", "right"]),
"apply_to_region" : probability_designation(),
...
def probability_designation(foot_view) :
    designation_num = random.random()
    if foot_view == "Indirect_plantar":
        if designation_num <= 0.206:
            return "heel"
        elif 0.206 < designation_num <= 0.256:
            return "mid_foot"
        else:
            return "upper_foot"
    else:
        if 0.754:
            return "upper_foot"
        else:
            return "mid_foot"
```

Excerpt of a wound's location parameters and region probability

While the foot selection is truly random, the specific location of the foot relies on a specific set of probabilities. The chances for each region to be selected have been inspired by the conducted literature review. Referring to the findings of Cowley *et al.* (2008) and their comprehensive table of real-world ulcer occurrences shown in Figure 1, the following probabilities have been integrated into FUERA's `probability_designation()` function:

1. Plantar: ~74.4% for the wound to appear in the upper foot, ~20.6% on the heel and ~5% in the mid foot.
2. Dorsal: ~75.4% for the wound to appear in the upper foot and ~24.6% on the mid foot. There is no percentage chance on the heel/ankle area, as the provided cropped arrays do not have ankles recorded - Figure 6 provides visualization.

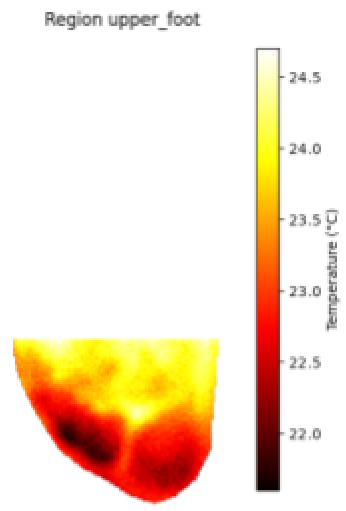


Figure 9: Region mask for the upper foot (other areas NaN)

The helper `_build_full_region_mask(heel, mid, upper, region)` then returns a full-height temperature array for exactly one of the desired regions, with the rest of the foot converted to `NaN` values. This enables FUERA to consider only the region of interest to apply the wound to, all while preserving the original geometric coordinates, foot shape and pixel dimensions of the image. Figure 9 visualizes one possibility for a region mask if the wound is to be placed in the upper foot.

Center selection. The center of the wound, denoted in the program by `(y_center, x_center)`, is determined by the helper function `select_center_and_shape_on_image()`. It utilizes the region mask and computes the wound center coordinates within that region by determining the valid pixels with `np.where(~np.isnan(region_full))`.

6.3.3. Wound Variables

Wound masks. Every wound in FUERA comprises two distinct masks, both applied together during the wound generation and development process. The *core mask* simulates the hot nucleus of a wound, and the *inflammation mask* represents the region of warmth around the wound core. Figure 10 shows how both are applied onto a healthy foot to synthesize a wounded patient.

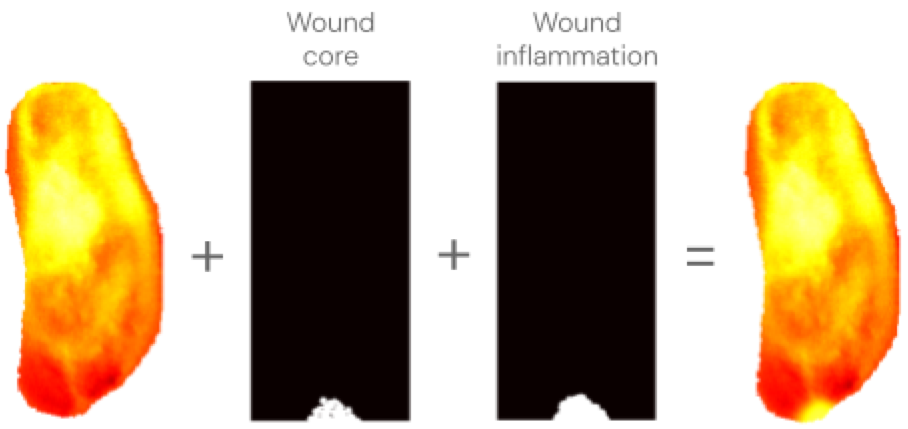


Figure 10: Wound mask application procedure

Shape selection. Each wound has two possibilities for its shape: "*circle*" and "*multi*". If the randomly chosen shape mode is a circle, then the function generates two coexistent circular core and inflammation masks around the predetermined `(y_center, x_center)` of the wound and radii `core_r` and `inflam_r` by using the equation to draw a solid circular disk, which includes the area inside the circle and, thanks to ≤, the boundary circumference:

$$X^2 + Y^2 \leq R^2,$$

and integrating the calculation to both masks in the circle mode:

```
core_mask   = (X - x_center)**2 + (Y - y_center)**2 <= core_r**2
inflam_mask = (X - x_center)**2 + (Y - y_center)**2 <= inflam_r**2
```

The resulting circle is plotted in one boolean mask that represents the high-temperature core of the generated wound and another boolean mask that represents the surrounding inflammation.

If the wound's shape is "multi", it generates a complex wound by essentially overlapping multiple smaller circular wounds (referred to in the context of a multi wound as *blobs*), and unioning them into one shape. A central anchor blob is placed at `(y_center, x_center)`, with additional blobs comprising the shape (amount determined and customizable via the wound parameters `"multi_min_blobs": 20` and `"multi_max_blobs": 50`). Every blob's generation is constrained to make them connected with one another by at least `MIN_UNIQUE_PIXELS = 3` pixels and at most `MAX_OVERLAP_PIXELS = 8` pixels, ensuring a naturally connected complex wound.

After all of the blobs have been placed, both the core and inflammation areas of every blob are unioned into one mask per area with `OR`. This avoids the temperature overlap stacking, had the blobs not been merged but rather applied on top of one another.

Size policies. Two different wound sizing modes are available to be switched to with the `SIZE_POLICY` flag. The *relative* policy is the default one, which calculates the inflammation and core radii by taking into consideration the dimensions of each of the foot arrays. For the core radius size, the program picks a small percentage of the temperature array from the range `CORE_RADIUS_FRAC_RANGE = (0.025, 0.040)`. The width and height of the foot canvas is thus multiplied by a percentage in the range of 2.5-4.0%, and those are made into the dimensions of the newly generated wound core. The inflammation radius follows, which is calculated as a multiplier of the core radius in the range of `INFLAM_OVER_CORE_RATIO = (1.6, 2.2)`. Moreover, since the wound area scales with R^2 (`inflam_r**2` in code), the inflammation area thus becomes 2.6-4.8 times the core area. These variables simulate the natural wound formation, with the smaller, hotter center and a larger, milder surrounding area.

The optional, less complex *absolute* policy is also possible, where the radii dimensions for both areas are picked between fixed, yet exclusive ranges:

```
ABS_CORE_RADIUS_RANGE = (8, 15)
ABS_INFLAM_RADIUS_RANGE = (16, 28)
```

Coverage limits. After mask construction, regardless of the size policy, a cap is enforced on the coverage of the wound. This acts as the final failsafe to ensure that the synthetic wounds are not formed unrealistically large relative to the foot it is generated on. The function computes how much each of the generated masks take up compared to the full region:

```
inflam_frac = inflam_area / region_area  # inflammation %
core_frac = core_area / region_area  # core %
```

If either of the masks exceed their respective predefined limits - >8% of the full non-`NaN` foot region for `inflam_frac` and >3% for `core_frac` - the function calculates the required rescaling needed to fit the masks within both allowed limits and scales both. More specifically, even if one mask fits its limit and the other one does not, both are still rescaled together to explicitly fit everything under either of the limits, while still preserving the relative (or absolute) dimension relationships between the core and inflammation areas.

Gaussian blur. The generated masks are sharp and flat, which undermines the realism of the generated wound. Therefore, Gaussian softening is applied to both after their generation by `gaussian_filter(wound_mask, sigma_mask)`, with the sigma (softening strength) value being in the range of 5.0 to 7.0. This blurs the sharp mask edges and makes the wound growth visually a part of the foot thermography.

6.3.4. Wound Progression

Wound progression is a pivotal stage of synthetic image generation. In the subsequent steps, the wound mask and its variables that were generated are then, over a set amount of time, transformed from zeros to their initialized values.

Timeline and progression modes. The synthetic wound timeline is split into three distinct phases of a hypothetical patient investigation: a period of wound absence (*healthy days*), a period of gradual wound growth (*development days*), and a period when the wound has reached its maximum capacity (*static days*). For the reasons of enhanced realism when generating a wound, FUERA employs two progression modes via `generate_weighted_random()`:

- *Standard progression* (60% chance): The timeline is fixed to 10 healthy days, 20 development days and 10 static days. Figure 11 demonstrates a sample of thermal images from a wound's development stage.
- *Variable progression* (40% chance): The healthy and static phases are still fixed to 10 days, but the number of development days varies within a range of 10-30. The bounds are customizable, and capture the natural per-person non-uniformity and speed of wound growth.

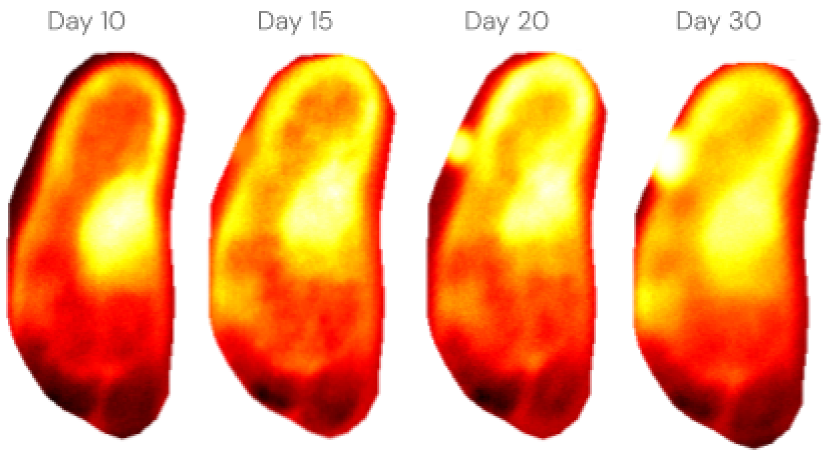


Figure 11: Development progression of a 20-day wound (after 10 days of initial temperature training)

```
def generate_weighted_random(20, 10, 30):
    # (standard_development, variable_lower_bound, variable_upper_bound)
    probability_roll = random.random()
    if probability_roll < 0.60:
        return x
    else:
        return random.randint(y, z)
```

The `generate_weighted_random` function (with its parameters fully customizable)

Daily progress value. Each day of the wound timeline is tied to a progress scalar `progress`, which drives the entire dual-mask growth and temperature increase during the development days. The initial healthy days have `progress` set to 0.0, the final static days have it set to 1.0, and during the development stage, `progress` is linearly increased from 0.0 to 1.0:

```
if day_idx < 10:  # 10 = healthy days
    current_phase = "healthy"
    progress = 0.0
elif day_idx < 10 + development_days:
    current_phase = "developing"
    progress = (day_idx - 10 + 1) / development_days
else:
    current_phase = "static"
    progress = 1.0
```

Wound variable development. During the development days, two fundamental variables of a wound are evolved by the program:

1. Intensity (or temperature) – From the first day of wound growth to the last, FUERA linearly increments temperatures of both the core and inflammation areas from 0 to `FINAL_INCREMENT_DEG_C` (3.0°C), in accordance with the current `progress` value:

```
increment = FINAL_INCREMENT_DEG_C * (progress if current_phase == "developing"
else 1.0)
core_target = core_base + increment
inflam_target = inflam_base + increment
```

2. Size – The radius of the wound's core and inflammation is linearly incremented over the development day period for that wound. The process begins with a minimum radius of the wound set by `initial_size_scale` (5%). As the wound evolves via `progress`, FUERA computes the daily current fractional size of the wound mask relative to its final dimensions and applies the scaling to both wound masks:

```
scale = initial_size_scale + (1.0 - initial_size_scale) * progress
core_mask = scale_mask(final_core_mask, scale, H, W)
inflam_mask = scale_mask(final_inflam_mask, scale, H, W)
```

6.4. Wound Detection Algorithm

6.4.1. Foot Grid & Chunking

Chunk definition. Each foot's thermal image, made up of pixels forming the temperature of the image, is partitioned into small, equally sized cells (*"chunks"*), each 5×5 pixels by default. The rationale behind grouping all pixels into such chunks is to reduce the excessive noise of the images, with 5px being a sufficient compromise between blurring the temperatures and algorithm's complexity. This is especially helpful on the foot that has been corrected in the previous stage, as discussed in Section 6.2, several transformations tend to create rough edges. This would cause false results to become more prominent, had the algorithm done per-pixel analysis.

NaN masking. FUERA interprets the chunk grid using the *display view* of each foot image for its processes. During the per-day data load, background chunks' zero values are converted to NaN, with this NaN-masked image of each foot utilized in all subsequent steps (grid building, statistics and visualization), ensuring that any background chunk will be filtered out by `np.isfinite`.

```
left  = left.astype(float);  left[left  <= 0] = np.nan
right = right.astype(float); right[right <= 0] = np.nan
```

How background chunks are detected and NaN-masked

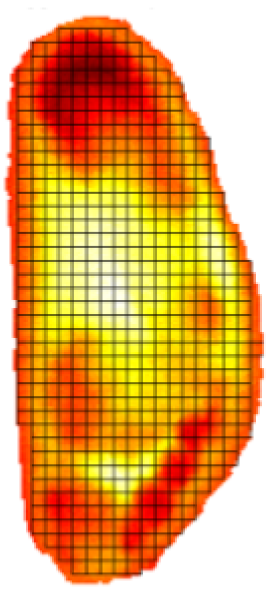


Figure 12:
Final foot grid

Grid generation. The preliminary grid framework is created by sliding a window in steps equal to the chunk size along both X and Y axes. This creates a grid that fits entirely within image bounds. This is not the final form of the grid however, as the foot grid used by subsequent steps of FUERA is constructed by taking all chunks which must adhere to one rule:

No NaN values in the display view. This ensures that each chunk sits fully within the foot boundary, covering as much of the foot area as possible without going outside of it. Moreover, this prevents the calculation of the mean chunk temperature from being skewed by 0-degree pixels that it would otherwise contain, thus representing true temperature values unaffected by the background.

The valid chunks form a perfectly fit foot grid, overlaid on each foot's geometry (see Figure 12).

Coordinate systems. There are two distinct coordinate systems utilized with each chunk, each with a distinct purpose. The *pixel coordinates* are the integer pixel positions on a foot canvas in (x, y)-form, with a chunk represented as `(x0, y0, w, h)`. The grid generation stage uses the pixel coordinates to lay down the chunks by using the relative `(x0, y0)` anchors - top left corner of each chunk - and moving across the canvas using the predefined `(w, h)` pixel dimensions. Moreover, this system is also used for grid visualization, including the grid lines, trigger/suspect chunks and wound clusters.

The *grid coordinates* is the standard lattice system of chunk indices. Each chunk has an index `(ix, iy)`, where `ix = x0 // chunk_px` and `iy = y0 // chunk_px`, and each step within the grid coordinates is exactly one chunk. This is not only useful for keeping chunk dictionaries (in the form of `{(ix,iy): (x0,y0,w,h)}`) and keying the sliding window, but also is an integral part of comparing specific chunks across the feet.

6.4.2. General Detection Strategy

FUERA employs two distinct detection methods to conclude its results for every case. The first and the one that is backed by earlier literature is the *contralateral check*, which means comparing the temperatures of the same anatomical regions of both feet. This is possible due to the symmetry of a human body, where both feet are essentially mirrored counterparts (most of the cases, but still requires noise control). As explained in its subsequent section, this check is not enough to avoid generating false positives, therefore a second additional check, the *unilateral check*, is incorporated. This looks at a single foot only, determining any chunk temperature deviations after undergoing a period of learning the thermal nature of the foot.

Chunk triggers. Thanks to the sliding window technique, the system has a clear detection progression integrated in each chunk. Figure 13 provides an overview of the three main detection states of FUERA.

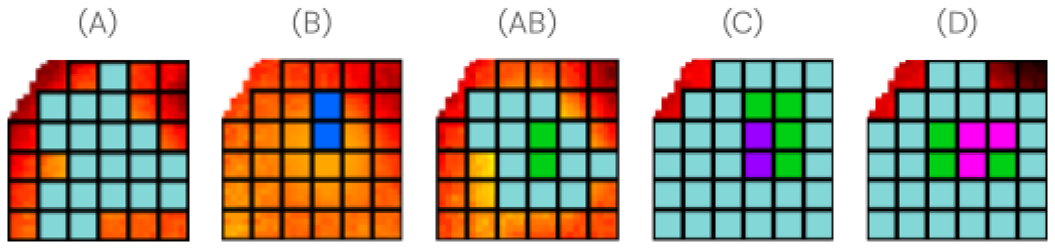


Figure 13: Chunk trigger progression on a hotspot

The first two stages are the baseline observations of per-tile temperature *trigger chunks*. Both contralateral and unilateral checks use different methods for detecting them, and are explained in their respective sections. In the display view, the trigger chunks are marked with cyan for the *unilateral triggers* (image A on Figure 13) and blue for *contralateral triggers* (image B on Figure 13). This is essentially a visualization of which chunks were entered as `1` in their history list on that day. Since it was decided to combine both checks into one detection pipeline, the subsequent stages utilize the *combined triggers* by taking `contralat_trig ∩ (unilat_trig_left ∪ unilat_trig_right)`, marking these chunks with green (image AB on Figure 13).

A chunk is then labeled as a *suspect chunk* if the sum of its combined triggers in the sliding window reaches `MIN_SUSPECT_TRIGGERS` (3), meaning that it has been suspiciously hot on 3 of the last 5 days. The suspect chunks are marked with purple color in the display view (image C in Figure 13).

FUERA reaches the final stage once three or more connected suspect chunks are detected, treating such formation as a *wound cluster*. The algorithm scans all chunks in grid space, and looks for any formed wound clusters in 4-connected connectivity: `neighbors = [(ix+1,iy), (ix-1,iy), (ix,iy+1), (ix,iy-1)]`. The day that a wound cluster appears on either foot, the outputs are saved and the detection cycle is stopped for that patient, marking the cluster chunks with magenta in the display view (image D in Figure 13).

AND for both checks. Both detection steps (contralateral and unilateral) were concluded to be crucial to the optimal performance of the algorithm. Initially, FUERA only had the former implemented, and having just the contralateral check consistently produced false positives following the 5/3 rule (initially the algorithm had a 10/7 rule in which the one check stably worked - that is 7 triggers per chunk out of the 10 days of the sliding window - however the client requested an quicker detection strategy with three triggers which deteriorated the contralateral-only system). As a result, the unilateral check was incorporated alongside it, with the algorithm working specifically via both. This reduces the false positives which flag the contralateral check earlier than expected by having the unilateral check as confirmation.

System logging. For the convenience of the patient and the doctor, the algorithm keeps a per-patient log file of the temperature abnormality, updating it after each analyzed day. The excerpt below demonstrates the status possibilities of the daily analysis, starting the log file after the learning period for the unilateral check has been completed. The log file also provides information on which foot and which region of it the suspect and cluster chunks are at.

```
...
Day 05: We're learning your normal temperature pattern. No concerns today.
Day 06: No suspicious hotspots today.
...
Day 26: We’re watching a warm area on your Right foot (upper-foot).
...
Day 29: We see signs that may indicate a wound on your Right foot
(upper-foot).
Detection finished. Please contact your clinician as soon as possible to
```

```
assess the area.
```

Excerpt from a `patient_log.txt` file

6.4.3. Contralateral Check

Grid traversal. Building directly on the grid structure established earlier, the contralateral check is done via the grid coordinate system, by comparing the same `(ix, iy)` position on the left and right foot. Such mirroring is possible due to the robust foot correction, as well as the fact that all provided temperature arrays have the same dimensions.

Sliding window. The sliding window functionality is the centerpiece of the algorithm's contralateral check, which modifies the framework day-by-day triggers into a persistent function. For each valid chunk `(ix, iy)` on each foot a short, rolling list of the most recent `WINDOW_DAYS` (5) days, meaning that all history lists have a maximum length of 5, naturally "sliding" through the timeline of the pipeline. Every day, an entry of a history list is `1` if that chunk was triggered on that day, else `0`.

Through testing and consultations with the client it was concluded that the pipeline employs a *5/3 rule* - 5 days of sliding window, and 3 triggers per chunk to mark suspicions. Therefore the variables mentioned in this section reflect this decision, but remain extracted at the top of the script for convenient modifications if needed.

```
# What the window is storing for each chunk -> {(ix, iy): 1 or 0}
window = defaultdict(lambda: deque(maxlen=WINDOW_DAYS))
...
window[ixiy].append(1 if ixiy in triggers_today else 0)
...
if sum(window[ixiy]) >= MIN_SUSPECT_TRIGGERS:  # MIN_SUSPECT_TRIGGERS = 3
    suspects.add(ixiy)
```

Sliding window main functionalities (simplified for the report)

Per-tile temperatures. The temperature of each chunk is calculated using the pixel coordinates, by going through the specific row and column pixel slices each chunk is made out of, by slicing the display views with `[y0:y0+h, x0:x0+w]` and finding their means with `nanmean`:

```
T_L = np.nanmean(left_display[y0:y0+h, x0:x0+w])
T_R = np.nanmean(right_display[y0:y0+h, x0:x0+w])
```

Following this step, the display views of each foot are refined into chunks with uniform inner temperatures, removing the overcomplicating pixel noise and preparing them for analysis with their contralateral counterparts.

Bilateral temperature difference. The pivotal idea of a contralateral check involves comparing the same chunks on both feet, in accordance with a certain threshold referenced in the established theory of ulcer detection (elaborated further in the Related Works section of the report). More specifically, the 2.2 °C degree contralateral difference has been established as a signal of abnormality and intervention. Therefore, FUERA incorporates this with an asymmetry threshold `temp_diff_threshold`, set to 2.2. The absolute difference of `T_L` and `T_R` is then calculated and compared with the threshold, and if the difference exceeds (or is equal to) 2.2, the algorithm records both chunks as triggered for that day.

6.4.4. Unilateral Check

In addition to the previously described method, as per client's requests to have a shorter detection period, a way to detect early ulcer detection using a single foot had been devised as an additional check. This allowed for the sliding window and trigger counter variables to be tighter, achieving the new requirements while alleviating incurred false positives. The main premise of this unilateral check is to detect early signs of wounds by comparing daily thermal images to a baseline of normal temperature patterns.

Training phase. This methodology includes a training phase wherein the system computes a baseline temperature profile by calculating the median values for each chunk over a period of 5 days. Although ten days of healthy foot images were provided for the project, only five were used to prevent overfitting and to allow for the possibility of realistically simulating false positives. Randomizing the training days ensured that the baseline captured general temperature behavior rather than being tailored to a specific period.

The baseline is meant to represent the normal temperature of each chunk of the foot. In addition to the median, the Mean Absolute Deviation (MAD) was also calculated for each chunk. The median was chosen over the `mean` because it is less sensitive to outliers, especially in smaller datasets. However, unlike standard deviation for mean, MAD is more resistant to outliers. As a result, a scaled version of MAD was utilized: `Scaled_MAD = 1.4826 x MAD`

Anomaly Detection. After the training phase, each new day's data is compared to the baseline for every chunk using the metrics given below:

- A chunk is marked as anomalous if its mean temperature deviates beyond the ±2 scaled MAD from the median represented in the formula: `∣ Tchunk − Medianchunk ∣ > 2 × Scaled_MADchunk`
- A chunk is also marked as anomalous if its mean temperature deviates beyond the range of values: `Tchunk < Tmin,chunk ∨ Tchunk > Tmax,chunk`

Detected anomalies are stored and tracked over a period of multiple days and if the anomalies persist for three or more consecutive days, then it is labelled as a wound.

7. System Testing

The project has two main deliverables: Generated synthetic thermal images of DFU's between stages 0 and 1, and FUERA - an early DFU detection algorithm.

Since it was not in the requirements specification to deliver a UI, a networked service, or a complete clinical product, many of the normal system testing expected for a final product are not within the scope of this project. Additionally, a fixed data set was used, so there are no risks in terms of runtime errors. Thus, the testing focused solely on the functional verification of the algorithm's ability to generate synthetic images. Essentially, verifying that the images generated follow the intended wound mask and that FUERA is able to detect the created ulcers.

Testing the synthetic image generator. The goal of this is to ensure the wound masks are applied correctly, that the temperature is adjusted accordingly, and these are maintained over the correct time period.

1. Mask application - Testing that the wound mask is created correctly
2. Temperature increase correctness - Tested that within the wound mask, temperatures increase by the correct amount, and any pixels not related to the wound mask were unaffected.
3. Wound progression - Tested mask application and temperature increase over multiple days to ensure the whole progression is correct
4. Gaussian Blur - Tested that the Gaussian blur that is applied preserves the mean wound temperature within a range of +-0.5°C thus it is not affecting the overall temperature and possibly detection
5. Integration tests - Tested that the .mat files are modified correctly (saving and loading) thus there are no rounding or other errors.

To summarize, all of the tests passed, thus confirming the system meets the requirements initially set out. The synthetic image generation properly creates and applies masks, and maintains temperature over multiple days, along with the pipeline of accessing and adjusting .mat files, being verified.

```
testGenerateHeatspotMask.py::TestSyntheticImageGenerator::test_parametrized_mu
lti_day_progression[7-3.0-1.0] PASSED [100%]

======================= 11 passed in 0.11s ========================
```

Excerpt from the testing terminal outputs

8. System Results & Validation

The previously stated tests, while useful, only tested whether the functions behaved as they were designed to be implemented. Actual validation and testing on unknown data for example could not be implemented due to the absence of ground truth of the characteristics of pre-ulcer diabetic foot wounds. Due to this, it was recommended that validation be done by using the client's subjective opinion of the results provided by the system. This form of validation, despite being far from perfect, was the sole validation criteria that could have been used to test the system results. The following two different approaches were used to validate the outcomes of the project's implementation:

1. Wound Generation Realism - Images with generated wounds were randomly selected and provided to the client. The client then provided a score from 1 to 5, with 1 being unrealistic and 5 being indistinguishable from the client's view on how a pre-ulcer diabetic wound could possibly appear to be.

2. Wound Progression Order - A sequence of images of the development of generated wounds were provided to the client with their order of progression jumbled. The client's task was to place the images in order of what was considered by him to be a realistic looking progression.

Wound Generation Realism. The results of the wound generation test of randomly selected images are displayed in the table below. The actual images used for the test can be seen in the appendix with the appropriate image label.

Image_Label	**Realism**
A	4
B	1
C	3
D	3
E	4

Wound Generation Realism Results

The model was given a 15 out of a possible 25 based on the client's view of how the wound could look like. The specific image that was lowly rated by the client (Image B) was a simple circular wound which was found to be too unnaturally shaped while the more complex wounds looked more indistinguishable compared to a possible real wound.

Wound Progression Order. The results of the wound progression order test are displayed below. The table contains the sequence labels (correct images can be seen in the appendix), the actual order of the sequence of wound generation, and the client's re-ordering of the sequence.

Sequence_Label	**Correct_Sequence**	**Client's sequence**
A	4-2-5-3-1	5-2-4-3-1
B	5-2-3-4-1	5-2-3-4-1
C	2-1-3-5-4	5-1-2-3-4
D	5-2-4-1-3	5-2-4-1-3

Wound Progression Order Results

Sequences A, B and D were almost perfectly aligned by the client (with a slight mismatch between 4 and 5 in Sequence A). The clear mismatch in Sequence C was due to the client mistaking a naturally warm spot on the foot with a wound.

9. Limitations & Future Research

Small dataset. The group was provided with a dataset of 150 healthy images consisting of 15 healthy patients over a period of 10 days. Additionally, a dataset of 30 images of ulcer patients were provided over a period of days. As mentioned before, several established techniques for wound generation and wound detection couldn't be integrated, such as GAN models, standard models, neural networks etc. However, once the product is deployed and begins capturing images from real patients, the dataset would be extensively supplemented, enabling the use of these more advanced techniques.

Limited theory. The sole, clinically validated metric available was the 2.2°C temperature asymmetry between contralateral foot regions, as established in prior literature. No peer-reviewed data existed on thermal signatures of pre-ulcerative progression, such as onset timing, spatial evolution, shape variability, or unilateral temperature anomalies. This theoretical vacuum forced heuristic assumptions in wound modeling (for example linear 10–30 day progression, probabilistic regional placement) and detection (unilateral anomaly thresholds based on median ± 2×scaled MAD over 5 training days). While these parameters were designed in consultation with the client, they remain theoretically unsubstantiated and potentially inaccurate, increasing the risk of false positives or missed early signals in real-world deployment.

Arbitrary Validation. Due to the absence of established theory and real pre-ulcerative thermal data, validation of the synthetic images and detection outputs relied entirely on subjective client feedback rather than rigorous scientific methods. Generated sequences were presented to the client for visual assessment of realism, wound placement plausibility, progression timing, and thermal pattern authenticity, with their responses being noted down. While this approach facilitated practical alignment with clinical expectations, it lacked objectivity, reproducibility, and statistical validity, being prone to individual bias and unrepresentative of broader expert consensus. No blinded reviews or quantitative clinical benchmarks were feasible, limiting the evaluation to anecdotal approval and underscoring a critical gap in formal validation.

10. Reflections

The project presented an opportunity to apply skills learned in previous computer science courses to supplement efforts to solve a very important problem, which is the early detection of diabetic foot wounds.

The process started with the collection of requirements directly from the client and then researching state-of-art methodologies to find their limitations. Due to the complex setup of the project, the unknown nature of diabetic foot wounds and limitations with the dataset, a proposal was presented to the client which consisted of the requirements, possible pitfalls, and methodologies to be researched and implemented, ensuring clear communication of what would be attempted. Early on it was decided that AI/ML methods couldn't be used which resulted in the group using an algorithmic approach for wound generation and detection. In fact, such direction was preferred by the client. The nature of model-based detection seemed, in client's words, too "black-box"-like.

The greatest challenges included generating a wound with a realistic appearance and aligning both feet to compare their contralateral regions. Regarding the generation, it was extremely difficult to have pin point accuracy for applying probabilities for particular areas, such as specific toes or the side of the foot. For aligning both feet, it was a very arduous process of testing and discovering which methods and strategies would work in preserving temperature data while warping the deformed foot image to make them as symmetrical as possible.

Results were shown to the client and the methodologies were explained. As the implementation appeared to be going on schedule, the client supplemented the list of requirements to include the unilateral foot check. It was decided to implement this in parallel with writing the report.

Results and validation presented its own list of problems. Despite the only available theory stating a 2.2 degree contralateral temperature difference between feet, the client in later weeks suggested not only following the theory, and additionally supplementing it by making assumptions on temperatures during wound formation. This had not been considered as it was assumed that the project had to follow scientific data to precision. Additionally, the non-medical background of the students involved also limited the ability to make scientific assumptions in such conditions.

Overall, this project brought many trials and tribulations as well as necessary lessons being learned ranging from how to work with a client, understand and handle their needs and wants and deliver a product that was beyond their expectations and achieved things requested that were beyond the initial scope, to working within a multi-cultural and multi-faceted team, learning to play to each other's strengths and complementing their weaknesses. The team also learned the value of communication and organization hand in hand with determination and perseverance. All in all, this project was a very pivotal experience for the whole team in general, and helped round and improve them holistically.

11. Individual Contributions

Team Member	Development	Report	Other contributions
Kurt Salapare	• Foot Segmentation • Primarily Synthetic Wound Generation (Probabilistic Wound Location, bug fixes for wound variables, generation for different wound types and overall generation) • Visualisation for bug fixes with Synthetic Wound Generation	• Synthetic Wound Generation • FInal Pipeline Diagram • Related Works Style Gan • Reflection Component	• Worked on making final presentation • Peer review presenter • Final presentation presenter
Dan Gladkov	• Processing .mat files strategy and visualization • Synthetic Wound Generation (Wound Location, Wound Variables, Wound Progression) • Wound Detection Algorithm (Foot grid and chunking, contralateral check, combination of two checks into one script, visualization)	• System Pipeline Overview • Data Familiarization + Visualization • Synthetic Wound Generation • Wound Detection Algorithm (Foot Grid & Chunking, General Detection Strategy, Contralateral Check)	• Final Poster Creator
Alex Duncan	• Foot Correction (Mirroring, Scaling, Rotating and Warping) • Implemented System testing (Unit tests for wound generation) • Wound Detection Algorithm (Assisted with determining a methodology for detection)	• Introduction • Problem Statement • System Pipeline Overview • Foot Correction (Mirroring, Scaling & Rotation) • System Testing	• Worked on making final presentation • Peer review presenter • Final presentation presenter
Narenda Setty	• Wound Detection Algorithm (Unilateral check) • Foot Correction (Warping)	• Requirements Specification • Related Work (Diabetic Foot Wound Locations, Early Detection, Limitations with state-of-art,	• Worked on making final presentation

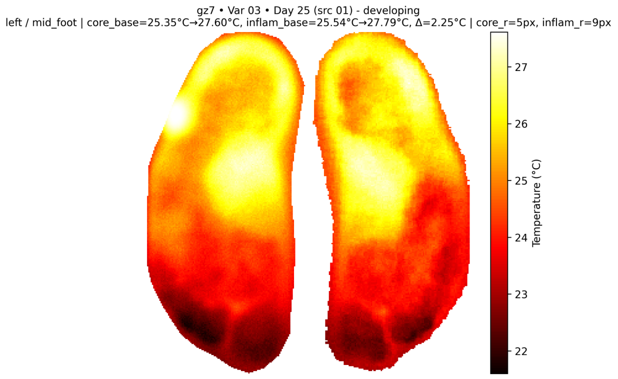
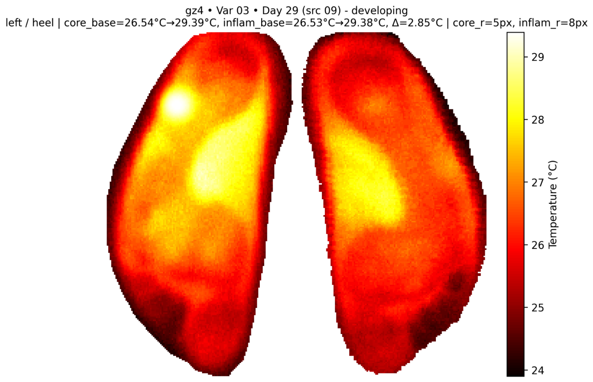
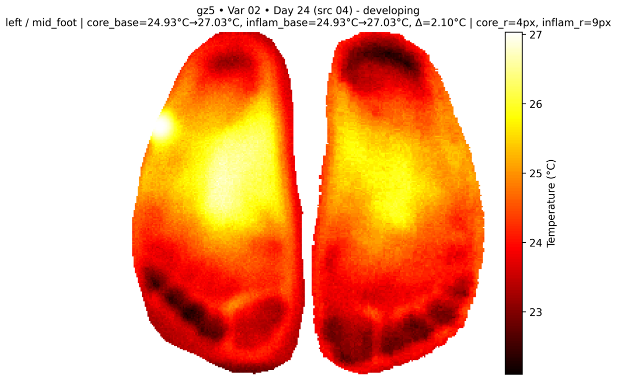
	•	Paper provided by client, Researcher Interview) • Foot Correction (Warping) • Wound Detection Algorithm (Unilateral Check) • System Results & Validation • Limitations & Future Research • Reflection	
Hoang Pham	• Wound Detection Algorithm	• Data Familiarization + Visualization	

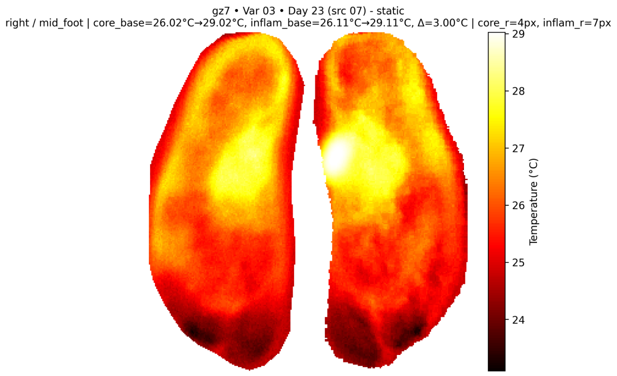
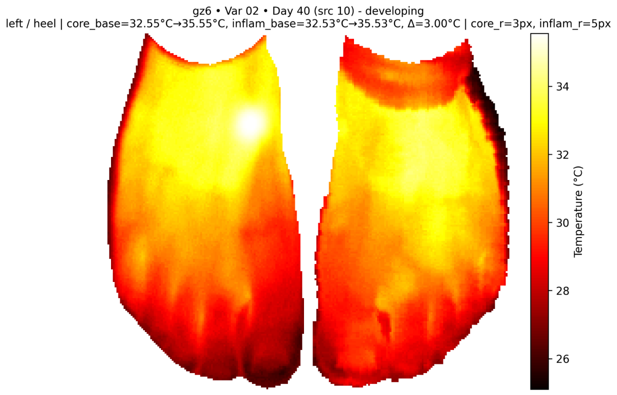
12. References

1. Brindha, S., Ramanakrishnan, A., Kirthika, R., & Monish Ram, V. S., (2024). Advancements in Diabetic Foot Ulcer Detection: A Thermal Imaging Approach. *2024 International Conference on Communication, Computing and Internet of Things (IC3IoT),* 1-5, https://doi.org/10.1109/IC3IoT60841.2024.10550286
2. Alshayeji, M. H., Sindhu, S. C., & Abed. S., (2023). Early detection of diabetic foot ulcers from thermal images using the bag of features technique. Biomedical Signal Processing and Control, 79(2), https://doi.org/10.1016/j.bspc.2022.104143
3. Zoetlief, E., (2023, August 23). *Explorative Study of Thermal Footprint Imaging as Method to Early Detect Diabetic Foot Ulcers in a Domestic Setting: the Bath Mat*. University of Twente. https://essay.utwente.nl/fileshare/file/96896/96896_Zoetelief_MA_TNW_TM.pdf
4. Lavery, L. A., Petersen, B. J., Linders, D. R., Bloom, J. D., Rothenberg, G. M., & Armstrong, D. G. (2019). Unilateral remote temperature monitoring to predict future ulceration for the diabetic foot in remission. BMJ Open Diabetes Research & Care, 7(1), e000696. https://doi.org/10.1136/bmjdrc-2019-000696
5. Cowley, M. S., Boyko, E. J., Shofer, J. B., Ahroni, J. H., & Ledoux, W. R., (2008). Foot ulcer risk and location in relation to prospective clinical assessment of foot shape and mobility among persons with diabetes. Diabetes Research and Clinical Practice, 82(2), 226-232, https://www.diabetesresearchclinicalpractice.com/article/S0168-8227(08)00380-X/abstract
6. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., & Aila, T. (2020). *Training Generative Adversarial Networks with Limited Data*. arXiv. https://arxiv.org/abs/2006.06676

13. Appendix

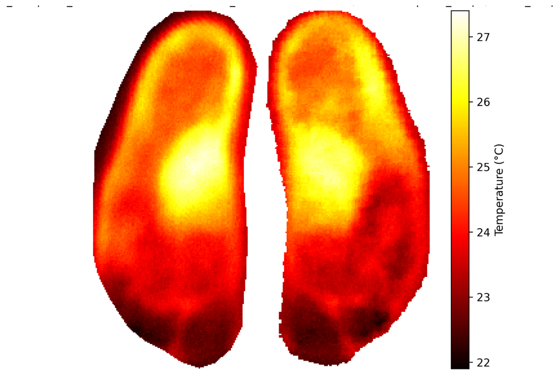
Wound Generation Realism:

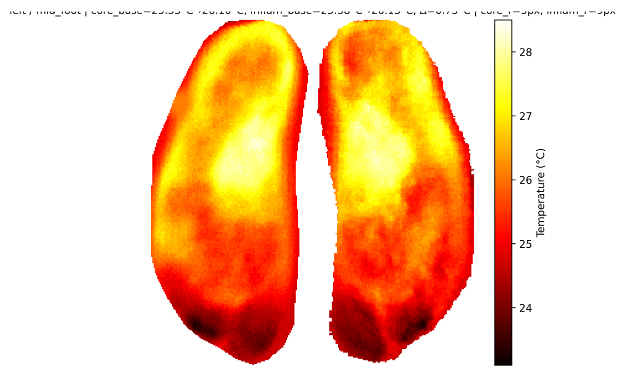
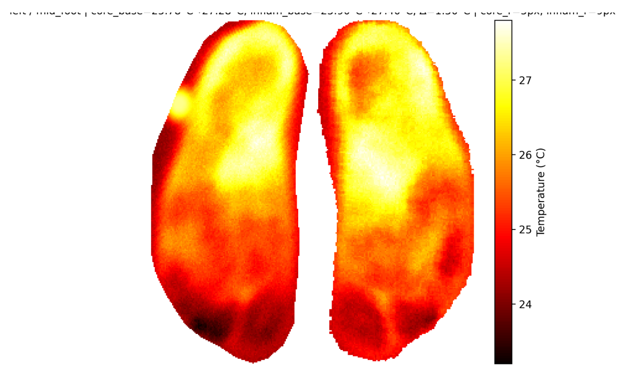
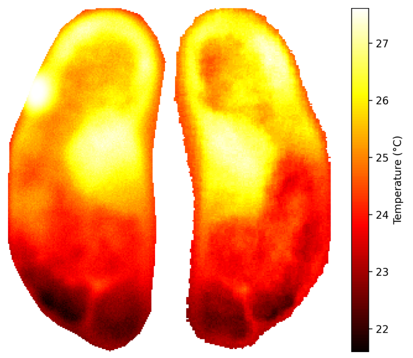
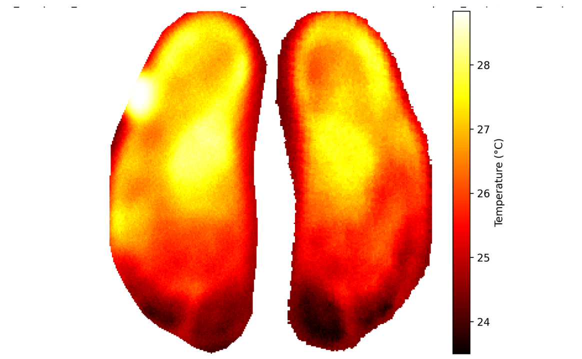
Label	Image
A	 gz7 • Var 03 • Day 25 (src 01) - developing left / mid_foot | core_base=25.35°C→27.60°C, inflam_base=25.54°C→27.79°C, Δ=2.25°C | core_r=5px, inflam_r=9px 27 26 25 24 23 22 Temperature (°C)
B	 gz4 • Var 03 • Day 29 (src 09) - developing left / heel | core_base=26.54°C→29.39°C, inflam_base=26.53°C→29.38°C, Δ=2.85°C | core_r=5px, inflam_r=8px 29 28 27 26 25 24 Temperature (°C)
C	 gz5 • Var 02 • Day 24 (src 04) - developing left / mid_foot | core_base=24.93°C→27.03°C, inflam_base=24.93°C→27.03°C, Δ=2.10°C | core_r=4px, inflam_r=9px 27 26 25 24 23 Temperature (°C)

D	 gz7 • Var 03 • Day 23 (src 07) - static right / mid_foot | core_base=26.02°C→29.02°C, inflam_base=26.11°C→29.11°C, Δ=3.00°C | core_r=4px, inflam_r=7px 29 28 27 26 25 24 Temperature (°C)
E	 gz6 • Var 02 • Day 40 (src 10) - developing left / heel | core_base=32.55°C→35.55°C, inflam_base=32.53°C→35.53°C, Δ=3.00°C | core_r=3px, inflam_r=5px 34 32 30 28 26 Temperature (°C)

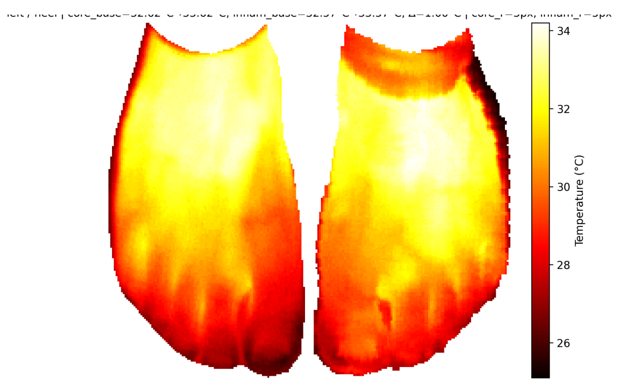
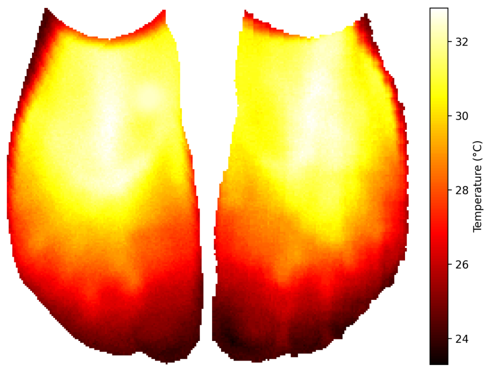
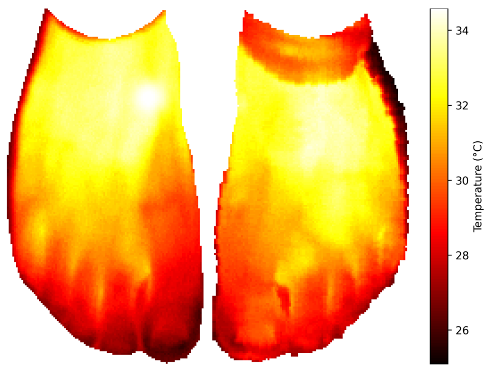
Wound Progression Order:

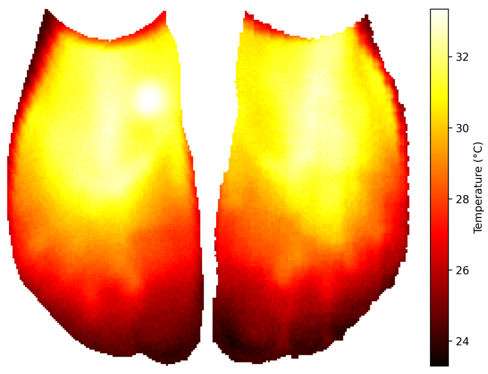
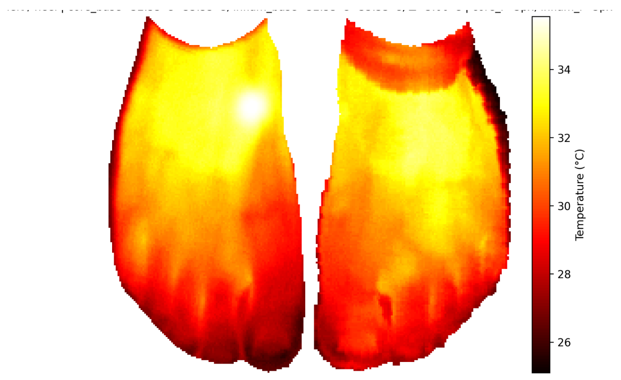
Sequence A

Image Order	Image
1	

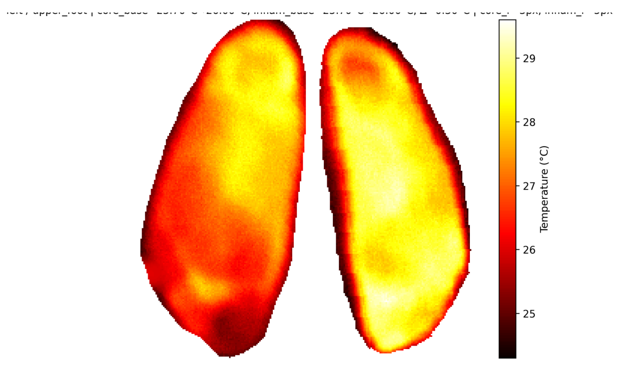
2	
3	
4	
5	

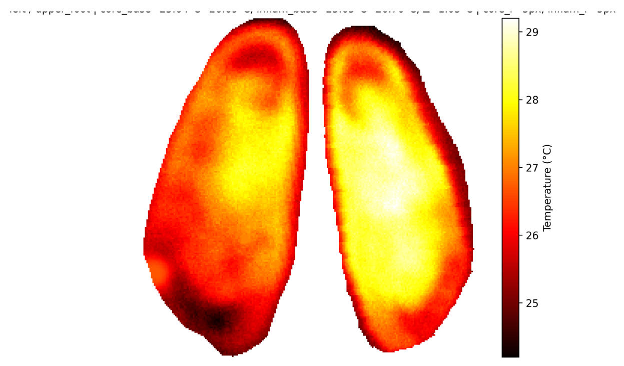
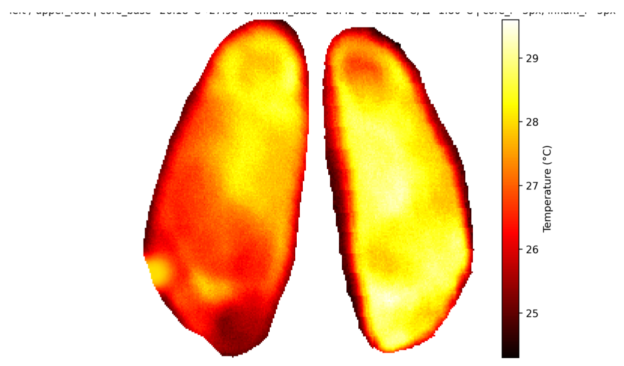
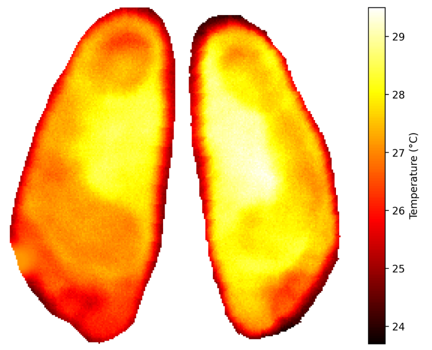
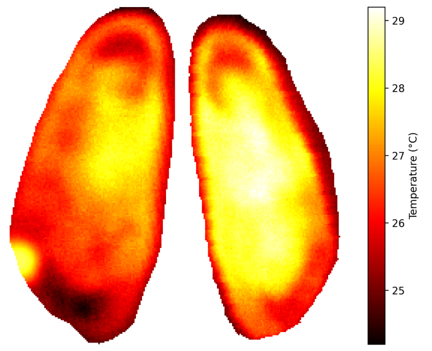
Sequence B

Image Order	Image
1	
2	
3	

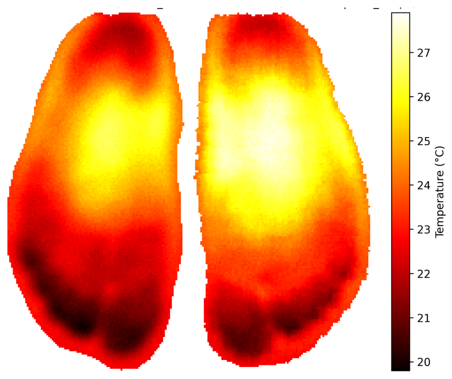
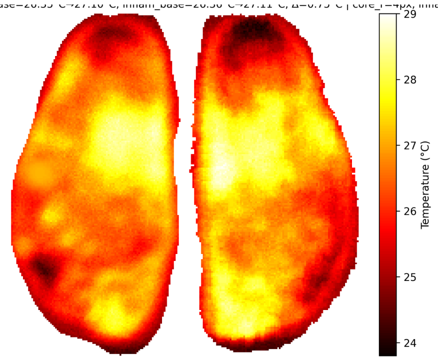
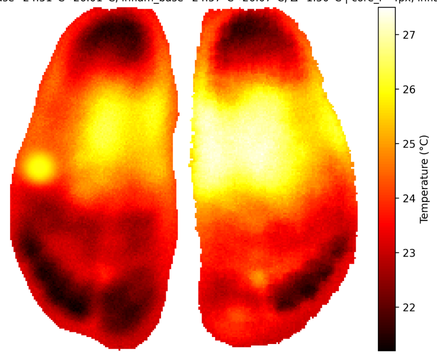
4	
5	

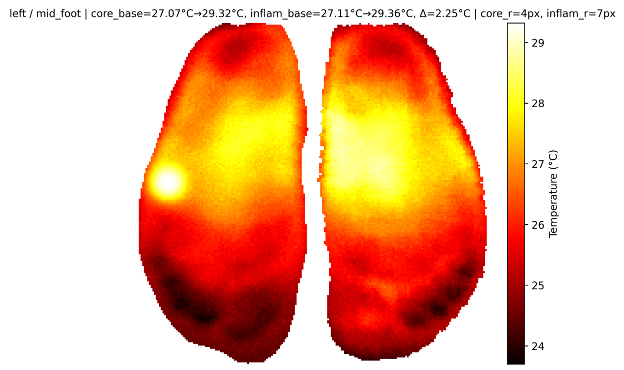
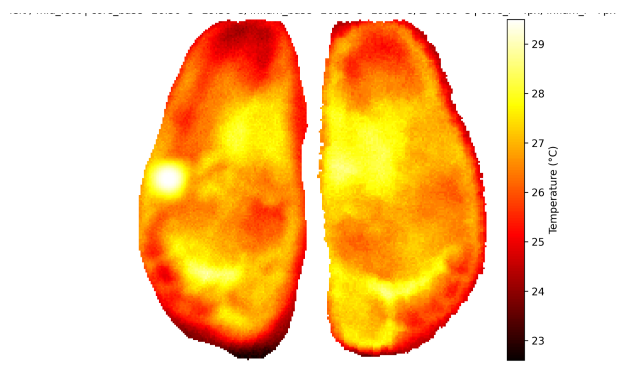
Sequence C

Image Order	Image
1	

2	
3	
4	
5	

Sequence D

Image Order	Image
1	 27 26 25 24 23 22 21 20 Temperature (°C)
2	
3	

4	left / mid_foot | core_base=27.07°C→29.32°C, inflam_base=27.11°C→29.36°C, Δ=2.25°C | core_r=4px, inflam_r=7px 29 28 27 26 25 24 Temperature (°C)
5	29 28 27 26 25 24 23 Temperature (°C)