1. Title:

   Problem: Cancer cell detection and preparing the model based on the dataset (Skin Cancer MNIST: HAM10000)

   About the Dataset: HAM10000 – Human Against Machine with 10000 training images

   1. Contains 10000 dermatoscopic images
   2. Also contains a metadata file (CSV) with demographics information of each lesion
   3. More than 50% of lesions are confirmed through histopathology
   4. The ground truth for the rest of the cases is either:
      a) Follow-up examination (follow_up) or
      b) Expert consensus (consensus) or
      c) Confirmation by in-vivo confocal microscopy (confocal)
2. Introduction: This design document describes the methodology for creating a cancer cell detection model. The objective is to create a model that, given a dataset, can correctly categorise cells as malignant or non-cancerous. The problem, the suggested solution, the rationale for the method selected, design specifics, the testing plan, and a discussion of potential future developments are all covered in great detail in this document.

3. Problem Description:

   Identifying cancer cells from a dataset of labelled instances is the task at hand. Obtaining a broad and representative dataset, preparing the data, extracting pertinent characteristics, choosing an acceptable model, training the model, assessing its performance, and deploying the model for predictions on new data are all necessary. Computational resources, time restrictions, and ethical considerations are a few examples of constraints. The labelled dataset serves as the input, while the trained model, which can categorise new cells as malignant or not, serves as the output.

4. Proposed Solution:

   The proposed solution involves the following steps:

   Dataset acquisition: Gather a diverse and representative dataset of labeled cancer and non-cancer cells.

   Data pre-processing: Clean the dataset, remove noise or outliers, and transform the data into a suitable format.

   Feature extraction: Extract relevant features from the data to differentiate between cancer and non-cancer cells.

   Model selection: Choose an appropriate machine learning or deep learning model for cancer cell detection.

   Model training: Train the selected model using the labeled dataset.

   Model evaluation: Assess the performance of the trained model using a testing dataset and relevant evaluation metrics.

5. Justification for Approach:

   Based on its compatibility with the criteria of the problem and the availability of labelled data, the chosen approach is justifiable. Convolutional neural networks (CNNs), one type of deep learning model, have demonstrated promising performance in image categorization applications including cancer cell detection. CNNs are a good choice for this issue since they can automatically extract pertinent features from the data. The method provides a fair compromise between precision and computational effectiveness.

6. Design Details:

   In the first step, I load the dataset. After downloading and unzipping the folder, I discover that it contains seven files. These files include two folders, each containing five thousand images, and an important file called HAM10000_metadata.csv. I begin by reading the metadata file and storing its contents in a pandas dataframe. Next, I add the images to the dataframe. To do this, I find the image path and define it as a new column in the dataframe. Then, I use this path to read the images. Additionally, I resize these images to 32x32 and convert them into NumPy arrays. This ensures that I am reading the correct image from the entire dataset. After this, I can check the value counts of each of the seven classes in the dataframe. Furthermore, I have the option to visualize a few samples to gain a better understanding of the data.

   Next, I proceed to handle the labels (dx) in our dataset. To accomplish this, I utilize the labelEncoder, which allows for the conversion of categorical labels into numerical format. Using the labelEncoder, I fit the labels and then transform them using the transform function. Subsequently, I incorporate the transformed labels as a new column called 'labels' in our dataframe. This addition enhances our ability to visualize and analyze the data distribution effectively. Upon plotting the data, it becomes evident that the dataset suffers from a significant class imbalance, as depicted by the graphical representation.

   In the next step, I focus on balancing the data. My goal is to ensure that each class has an equal representation in the dataset. To achieve this, I employ a technique where I select five hundred images from each class. In cases where a class does not have enough data to meet this threshold, I augment the dataset by adding replicative images randomly. To facilitate this process, I create separate dataframes for each class, allowing me to augment them individually. After augmentation, I concatenate these dataframes to form a new balanced dataframe.

   Moving forward, I follow the same instructions as in the first step. Additionally, since it is a multi-class classification task, I convert the labels into categorical format using one-hot encoding. This encoding scheme facilitates the representation of multiple classes in a suitable format for classification purposes.

   In the final step, I proceed to split the dataset into training and testing sets. I allocate 75% of the data for training and reserve the remaining portion for testing purposes. This division ensures that we have sufficient data to train our model effectively.

   Next, I define the model using Autokeras, an automated machine learning library. In our code, Autokeras explores twenty-five different models and selects the best one based on its performance. This allows us to proceed with the most promising model for further steps.

   Subsequently, I train the model on the training data. During this process, I iteratively evaluate the model's performance by adjusting the number of epochs. Through trial and error, I seek to find the optimal number of epochs that maximizes accuracy and ensures effective learning. By doing so I got the accuracy of 0.802285.

7. Conclusion:

   - This design document outlines the approach for building a cancer cell detection model. By acquiring a labeled dataset, preprocessing the data, extracting relevant features, selecting and training an appropriate model, and evaluating its performance, we can develop a robust cancer cell detection system. Continuous monitoring and maintenance of the deployed model are essential to ensure its accuracy and reliability.
   - Future improvements may include incorporating more advanced deep learning techniques, exploring transfer learning approaches, and expanding the dataset for better generalization.