Skin Lesion Segmentation and Classification

COSC 6370 - MEDICAL IMAGING

Project Proposal by:

Sri Lakshmi Meghana Yalamarty, 2028862, Computer Science, Graduate, smanepal@cougarnet.uh.edu

Divyasri Padala, 1991418, Computer Science, Graduate, dpadala@cougarnet.uh.edu

Viswanath Puppala, 2084700, Computer Science, Graduate, vpuppal2@cougarnet.uh.edu

INTRODUCTION:

Melanoma is a type of cancer which affects the skin (mostly in pigment cells) and grows very quickly if left untreated. It can spread to the lower part of our skin (dermis) enter the blood stream and then spread to other parts of the body. Melanoma that occurs on the skin called cutaneous melanoma, is the most common type of melanoma. It is reported that approximately 9,500 people in the U.S. are diagnosed with skin cancer every day and estimated that 197,700 new cases of melanoma, 97,920 noninvasive (in situ) and 99,780 invasive, will be diagnosed in the U.S. in 2022.

Melanoma, if detected in its early stages of growth, is highly curable. However, it is visually challenging to differentiate between melanoma (Cancerous) and benign skin lesions (noncancerous and often harmless) due to their similarities in appearance, making it difficult to detect even for trained medical experts. Hence, an intelligent medical imaging-based skin lesion diagnosis system can be very useful to identify and classify skin lesions faster and with more accuracy. This helps the physician to quickly deliver the needed medical attention to required patients thereby increasing their chances of survival.

In this project, a Convolutional Neural Networks (CNN's) for accurate skin lesion segmentation using U-net algorithm is proposed. It is a combination of Deconvolutional network and Fully Connected Network (FCN) that is specially designed for biomedical image segmentation. Further the ROI is processed to collect a series of features, scanning the skin area of interest for asymmetry (A), border irregularity (B), colors variegation (C), diameter (D) and texture (T) using efficient ABCDT feature extraction technique. Then the extracted features are fed into the Support Vector Machine (SVM) classifier for classification.

OBJECTIVES:

1) **Image Segmentation:** Simulate segmentation masks similar to the manual ground truth masks created by physicians to identify the region of skin lesion using Convolutional Neural Networks. This would make the process of skin lesion identification quicker when compared to manual processing.
2) **Feature Extraction:** Derive attributes from the segmentation masks to understand the distinct features representing a skin lesion, that allow in classification of a lesion as cancerous or non-cancerous.
3) **Image Classification:** Perform classification of images as Benign or Cancerous based on the features derived from segmentation masks.
4) **Image Pre-processing:** Perform image pre-processing before predicting the segmentation masks to ensure any noise due to hair on skin is eliminated for better prediction of segmentation masks.
5) **Graphical User Interface:** Design and implement a GUI that visualizes the pre-processing steps done and the results of the segmentation, classification done on skin lesion images.

LITERATURE REVIEW:

In 2009, a study was done by Alcón et al. that proposes an algorithm to detect malignant melanoma from benign lesions by the usage of skin lesion macroscopic images. In this study, for lesion area segmentation, first the elimination of the low frequency spatial component of the image was used for background correction, and then a thresholding-based method inspired by Otsu's algorithm, was used to segment the skin lesion area. Further using ABCD criteria, 55 features were extracted from the determined lesion area. Then correlation-based feature selection method and AdaBoost classifier were used as a feature selection step. In this algorithm, one decision support part was added which led to the usage of the personal information including skin type, age, gender and part of the body along with the output of image classifier. Finally, 86% accuracy, 94% sensitivity and 68% specificity have been achieved. [1]

Sohaib Najat Hasan et al. proposed a model to perform linear filtering and image restoration to obtain images free of hair and other artifacts. The pre-processed images were then passed to a modified U-Net architecture of 46-layers. [2]

In 2016, at the International Skin Imaging Challenge (ISIC) different methods were proposed and submitted for skin cancer segmentation, feature extraction, and classification tasks. The final report published a comparative study and showed significantly higher segmentation and classification accuracies of 95.3% and 91.6% respectively. The classification results were based on recognizing only two types of cancer, benign and malignant. [3]

The database of the mentioned studies was limited due to the conditions and constraints. This disadvantage prevents the proposed procedures from being appropriate to be applied on publicly available equipment that is the goal of proposing these procedures. Usage of personal information is not recommended to distinguish which also leads to non-uniform results. Finally, added layers in the U-NET architecture leads to higher processing times without much difference in segmentation performance.

DATASET:

For our current project we'll be using the HAM10000 dataset [4] consisting of the dermatoscopic images of common pigmented skin lesions. This dataset is taken from Harvard Dataverse and comprises of 10015 skin lesion images of seven classes: Actinic keratoses and intraepithelial carcinoma / Bowen's disease (akiec), basal cell carcinoma (bcc), benign keratosis-like lesions (bkl), dermatofibroma (df), melanoma (mel), melanocytic nevi (nv) and vascular lesions (vasc). From the existing 7 categories of skin lesions the bkl, df, nv, vasc are benign and akiec, mel, bcc are cancerous lesions.

Files in the dataset:

- HAM10000_images_part1 – Contains 5000 skin lesion images
- HAM10000_images_part2 – Contains 5015 skin lesion images
- HAM10000_metadata – Contains the metadata about all the 10015 images in the dataset
- HAM10000_segmentations_lesion_tschandl – Contains the Ground Truth segmentation masks for all the 10015 images

METHODOLOGY:

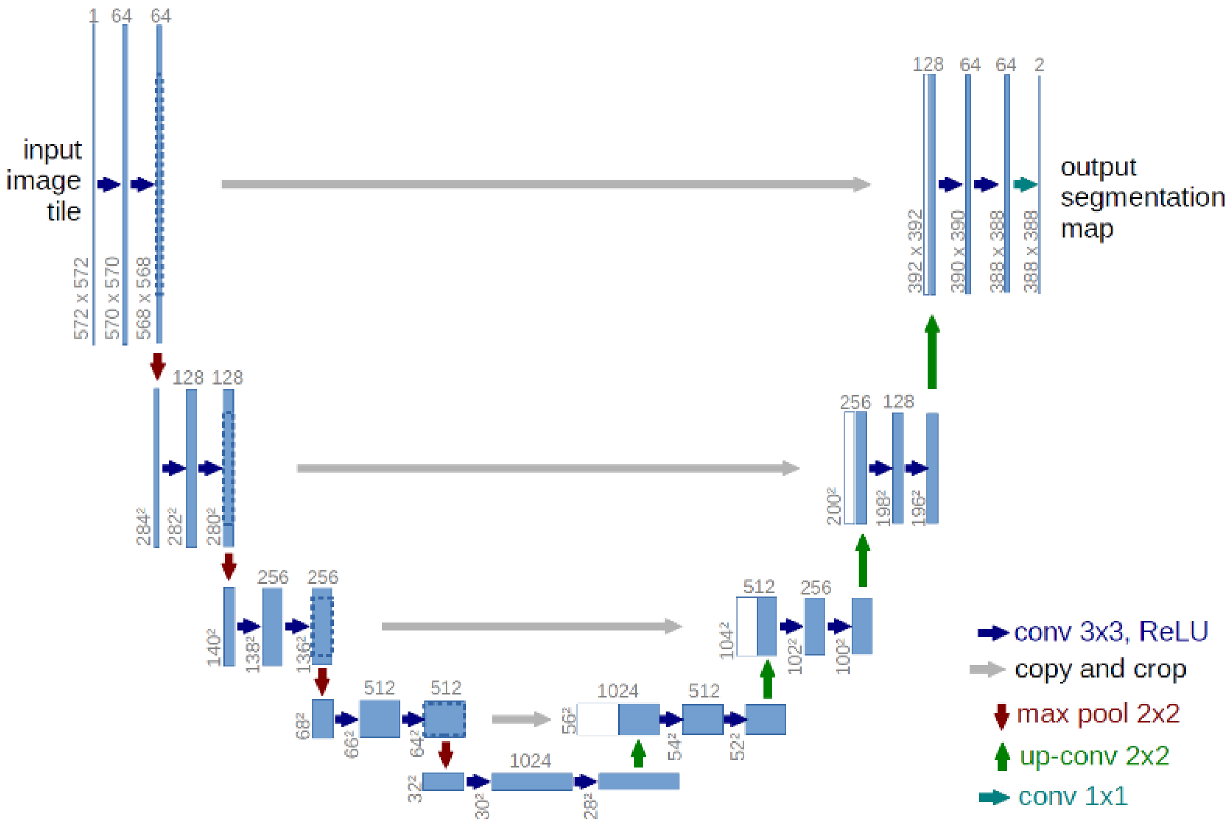
The algorithms/ methodology used for the implementation of our project are as described in the sub-sections below:

A) Image Segmentation:

- In this step we try to achieve our first objective by implementing the UNET algorithm to create the segmentation masks of skin lesion images that learn from the ground truth images received from the dataset.
- UNET Algorithm:

  The architecture can be divided into 2 parts, commonly known as contracting and expansive paths.

In the figure below, each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. [6]



UNET Architecture

Contracting (left side):

- Here, only a valid part of convolution is used; that is for every 3X3 convolutions, a 1-pixel border is lost. This allows for the processing of large images in individual tiles later. Each convolution is followed by a ReLU and batch normalization.
- To reduce the spatial dimensions, a 2X2 max pooling operation uses the xy size of the feature map. It propagates the maximum value i.e., the most activated pixel from each patch/pool (if filter size =2, then it is a 2X2 window pool) to the next feature map. After each max pooling operation, the results are down sampled, increasing the number of feature channels by the factor of stride (=2) while cutting in the spatial dimensions.

Expansion Path (right side):

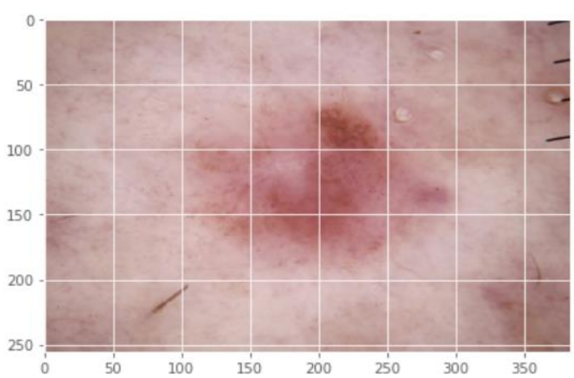
- This Expansion task is a series of up-convolutions and concatenation with high-resolution features from the contracting path.
- The 2X2 pixel output window is mapped to each feature vector using this up-convolution. Again, following the activation normalization, resulting a feature map that have factor 2 higher resolution decreasing the number of feature channels.
- It is also possible to combine the corresponding feature map from the contracting path, often by a 3X3 convolution.
- Finally, a 1x1 convolution is used to map the channels to the required number of classes.

- To implement the above described UNET algorithm for our dataset, we perform a train test split on the complete dataset of 10015 images. The split is done using the train_test_split() function from the sklearn.model_selection package.
- Observing the distribution of classes in the dataset we see that most of the images belong to the class 'nv' as shown below

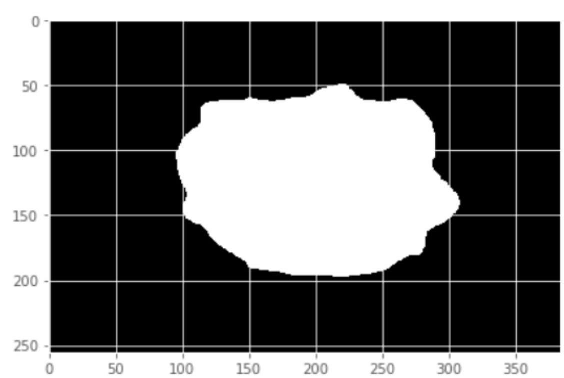
```
dx
akiec     327
bcc       514
bkl      1099
df        115
mel      1113
nv       6705
vasc      142
```

- Due to this imbalance, we do a train_test_split of the dataset by stratifying on the 'dx' column of the metadata. This ensures that there's equal distribution of the categories among the training and testing datasets.
- Next, we create two numpy arrays for the input images as X and for the target masks as Y for the data in the training set.
- Further, from the training set we split it into training and validation set to be used while evaluating our UNET model during its training.
- Since, UNET algorithm is based on convoluted neural networks which are used for supervised learning we'll need to give our algorithm the input images and their corresponding masks during the model training. The sample inputs given to our UNET model are as follows:

INPUT X for UNET model:



TARGET Y for UNET model:



- In our current implementation, the UNET model has been defined in the function `unet()` present in the `HAM10000_training.ipynb` file that defines the layers in the model's contracting and expansive blocks.
- We have also implemented Batch Normalization between the layers of the UNET architecture to speed up the training of our model.
- The functions used to perform convolution, MaxPooling, Concatenation and Dropout operations in our UNET model have been imported from the keras package in python.
- After defining the model, we initialize it with the dropout rate and the number of neurons to be used for the model training.

- The model is then used to fit() the training dataset that we created to learn from the input segmentation masks. We train our model for 10 Epochs with a learning rate of 0.001 and receive a final accuracy of 92%.

```
model.evaluate(X_test, Y_test, verbose=1)
```

```
13/13 [==============================] - 36s 3s/step - loss: 0.2371 - accuracy: 0.9188
```

- One limitation we encountered while training our model is due to the restricted hardware resources on our laptops. The complete training set of size 7511 could not be used as it was taking very long to train the model. We used a subset of 2000 images instead and were able to train our network in around 3.5 hours.
- Below are the results of the masks predicted by our trained UNET model

OBJECTIVE 1 RESULTS:

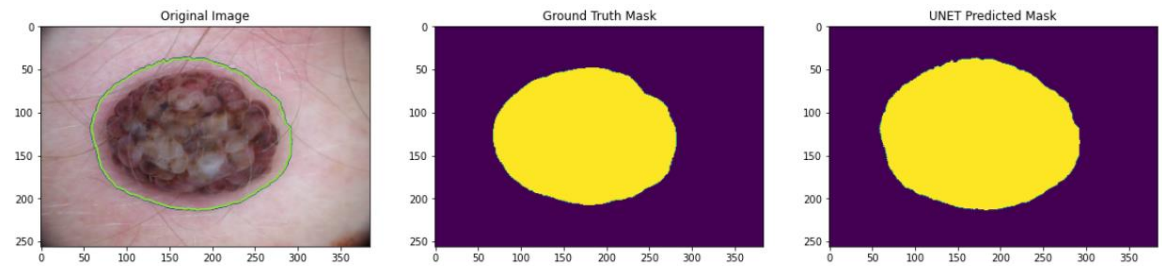


Figure 1

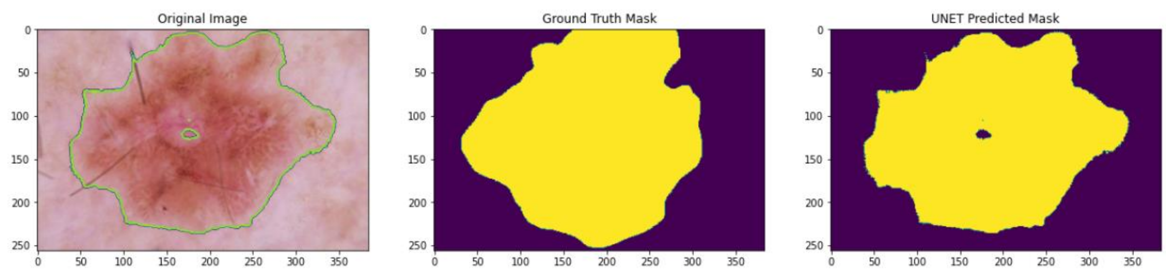


Figure 2

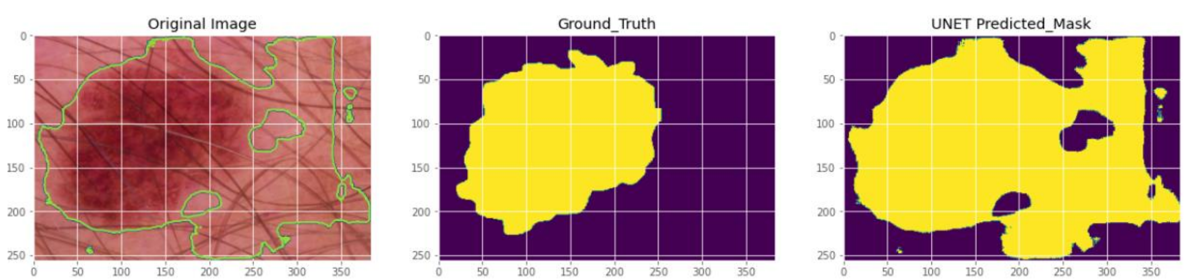


Figure 3

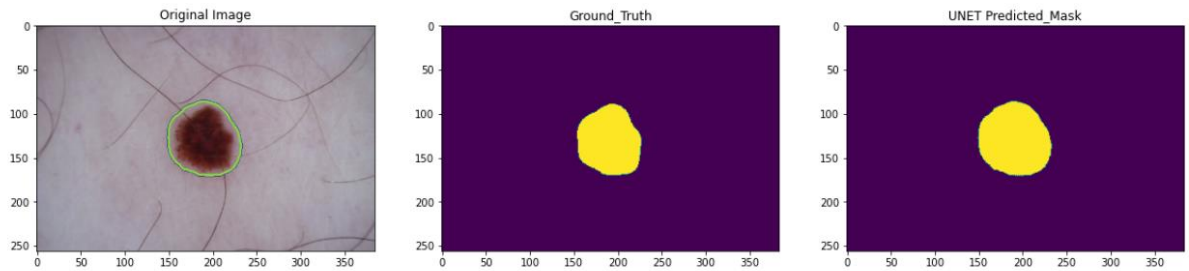


Figure 4

- From the above results we see that the model is able to predict the masks quite well if there isn't much noise due to hair as in Figure 3. So, to overcome this we tried out two options:
    i) Retraining our UNET model on preprocessed images
    ii) Send preprocessed images to our UNET model as inputs for prediction

  With approach 1, we ended up overfitting our model that started drawing masks for any small imperfection in the skin lesion images resulting in multiple small masks for a single lesion image being drawn or underrepresenting the lesion area in the image due to quality being lost during preprocessing. Hence, we chose approach 2 to train our UNET model on the original HAM10000 dataset and later preprocess images while using it for prediction.

B) Feature Extraction

- In the second step of our project implementation, we extract features from the above predicted segmentation masks to enable classification of the images in our next step. The implementation of this step achieves our objective 2 stated initially.
- In the everyday process of analyzing skin lesions by physicians they perform the classification of lesions by studying the properties of lesions such as its Asymmetry, Border irregularities, Color variegation, Diameter and Texture. We adopt the same process to define features to the images in the HAM10000 dataset.
- For every image in the training and the testing datasets we derive the below properties [5] from the image segmentation masks we calculated in the previous step.
    i) Asymmetry:
    In our current implementation, we represent the asymmetry of a skin lesion by its Asymmetric Index (AI) and Eccentricity. AI is defined as the average of the differences between the skin lesion with its horizontal flip and vertical flip. Eccentricity is used to understand how flat, or round is the ellipse in our mask.
    ii) Border irregularity:
    Border irregularity is represented by using the compact index which is calculated with the below formula, where P = Perimeter of lesion and A = lesion area

$$CI = \frac{P^2}{4\pi A}$$

    iii) Color Variegation:
    Color variation in a skin lesion is calculated by the standard deviation of the red, green and blue components of the image from their Maximum values in an image.

    iv) Diameter:
    Diameter is calculated for a circle with the same area as that of the skin lesion.

v) *Texture:*
The texture of an image is characterized by its consistency in the spread of the colors and patterns. In our current implementation we calculate texture by using the Grey Level Co-occurrence Matrix (GLCM) from the skimage.feature module.

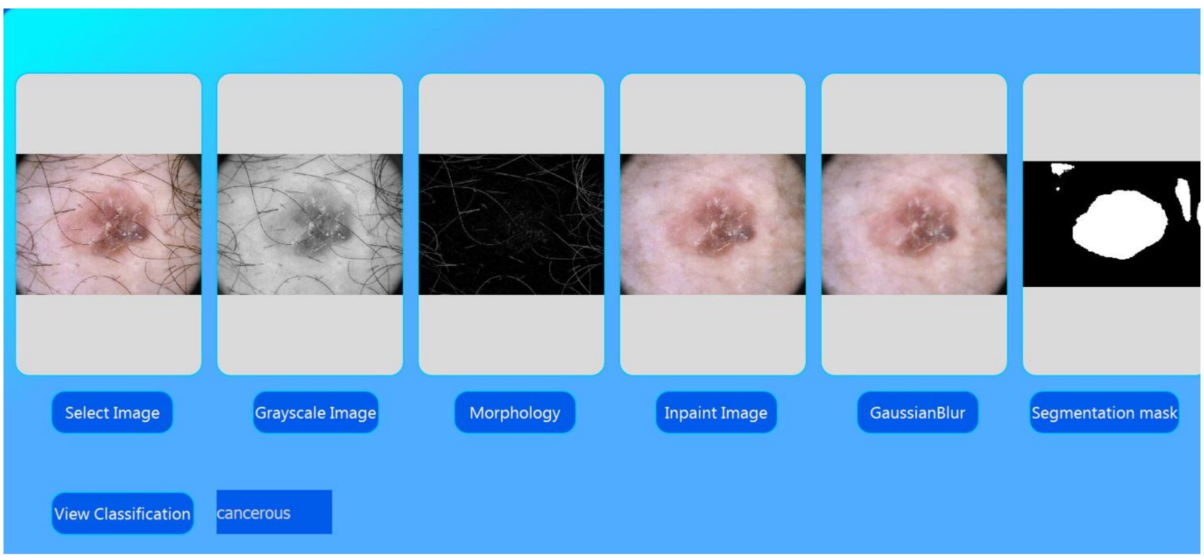
- All the above discussed methods are implemented in the `extract_features()` function in our `HAM10000_training.ipynb` file.

C) Image Classification

- To achieve our third objective, we take the features calculated in the above step for the training and testing datasets to train a SVM model with a Polynomial Kernel of degree five.
- The extracted features are given as an input to the SVM model training along with the target variable Y which represents if the image is Benign or Cancerous.
- The resulting model has an accuracy of 80% on the test dataset as shown below:

```
Accuracy: 0.804
F1-score: 0.716
Recall: 0.804
Precision: 0.646
```

OBJECTIVE 3 RESULTS:



We see that the model has predicted the image as cancerous which is according to it actual Y label "cancerous".

- One limitation for the SVM model we created is due to the data imbalance for the Cancerous vs Benign classes. As seen below cancerous class constitutes to only 19.5% of the whole dataset. This has led to the classification accuracy of the "Cancerous" class not being too good.

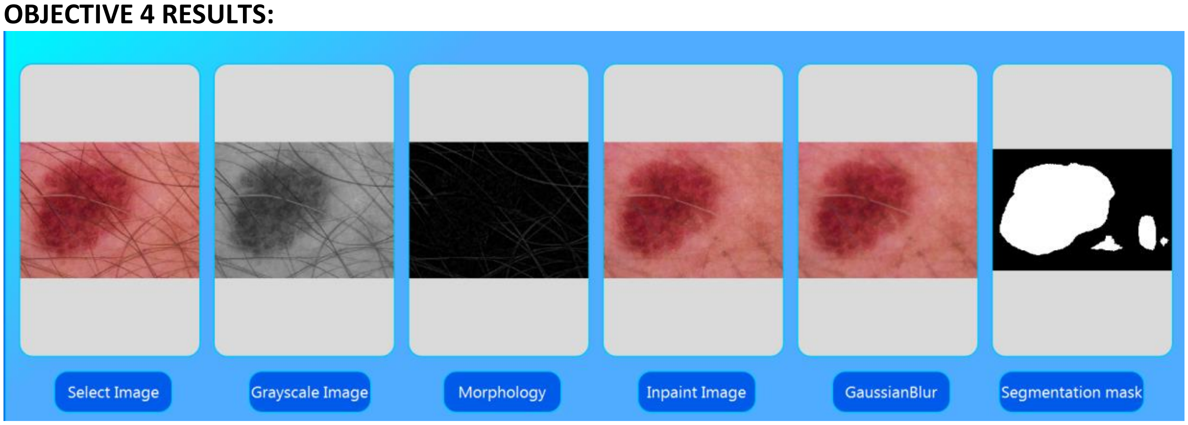
```
dx
benign       8061
cancerous    1954
```

- For future scope we plan to train the classification model by giving class weights or by performing data augmentation for the cancerous class to improve the model classification accuracy for this class.

D) Image Pre-processing

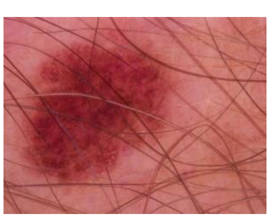
- In this step we perform pre-processing of the test images before they are given as input to our UNET model. This increases the accuracy of the segmentation results of the skin lesion image. The following steps have been performed to achieve this objective.
  - *i) Grayscale Conversion:*
    It is the process of converting colored images (RGB) to gray shades varying between complete black and white. Here, for eliminating the hue and saturation information, we are converting RGB image to gray scale using cv2.cvtColor() method from OpenCV library.
  - *ii) Morphology:*
    Morphology is an image processing technique that performs simple transformations using a structuring element called a kernel. In our implementation we are defining a kernel of size 17*17 pixels and performing a blackhat operation on the grayscale image. This is done to find the hair contours in an image with the help of cv2.morphologyEx() function.
  - *iii) Inpainting:*
    Inpainting is a technique used to reconstruct a selected image area. We use this technique to eliminate any hair follicles in the image by replacing it with pixels surrounding the region. This operation is carried out using the cv2.inpaint() function using the INPAINT_TELEA algorithm. The INPAINT_TELEA is based on a Fast Marching Method that performs inpainting by replacing a pixel with a normalized weighted sum of all the surrounding pixels.
  - iv) *Gaussian Blur:*
    Gaussian blur is a smoothing technique that is very useful in filtering out the gaussian noise from an image. We use cv2.GaussianBlur() to smooth our image before passing it to our UNET model for segmentation mask generation.

OBJECTIVE 4 RESULTS:

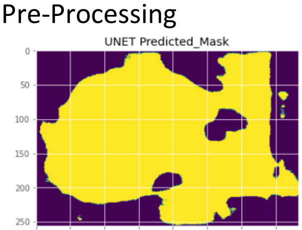


We can observe that for the same image referred to in Figure 3 of objective 1 results, the segmentation mask created after pre-processing image is so much nearer to the actual ground truth image than before pre-processing it. The same can be visualized below:

Actual Image



Segmentation Mask before Pre-Processing



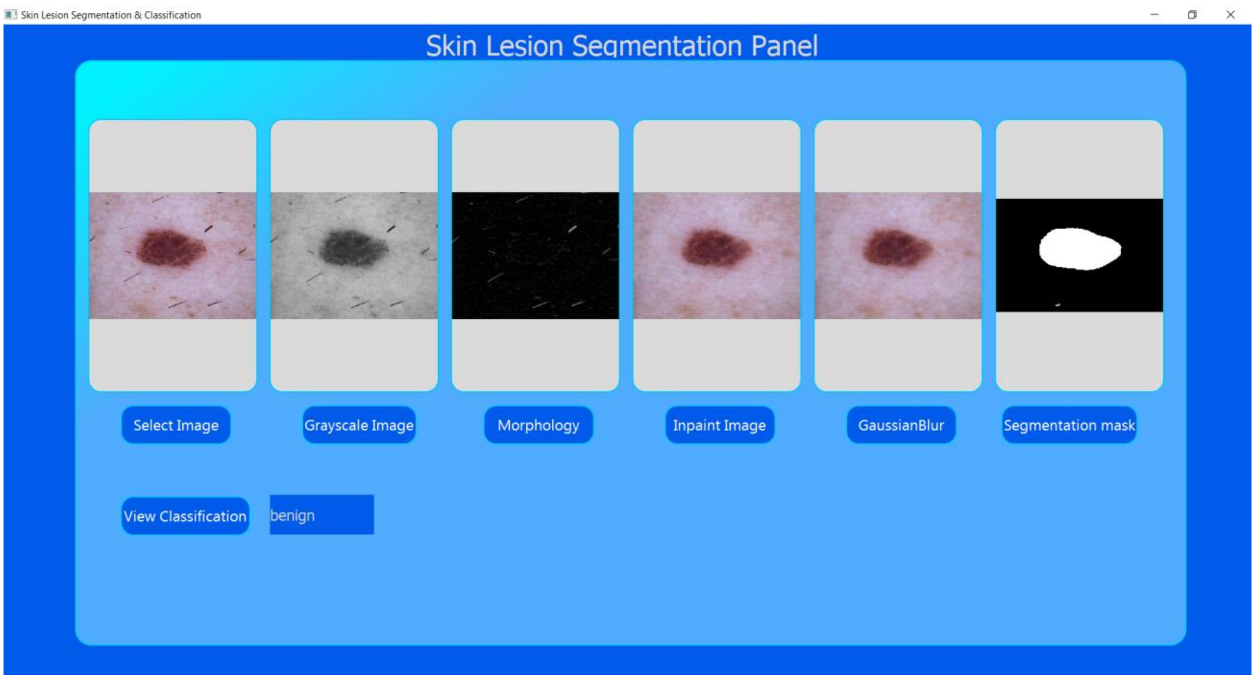
Segmentation Mask after Pre-Processing

Ground Truth Mask

E) Graphical User Interface

- The final objective of our project is to implement a GUI for enabling more intuitive access to the application. In our current implementation we developed a basic GUI using PyQT5 which enables a user to select an image and click on the buttons sequentially to perform the data pre-processing steps, image segmentation and classification of the skin lesion images.
- The GUI only allows sequential execution of the steps (buttons) shown on the GUI from left to right.

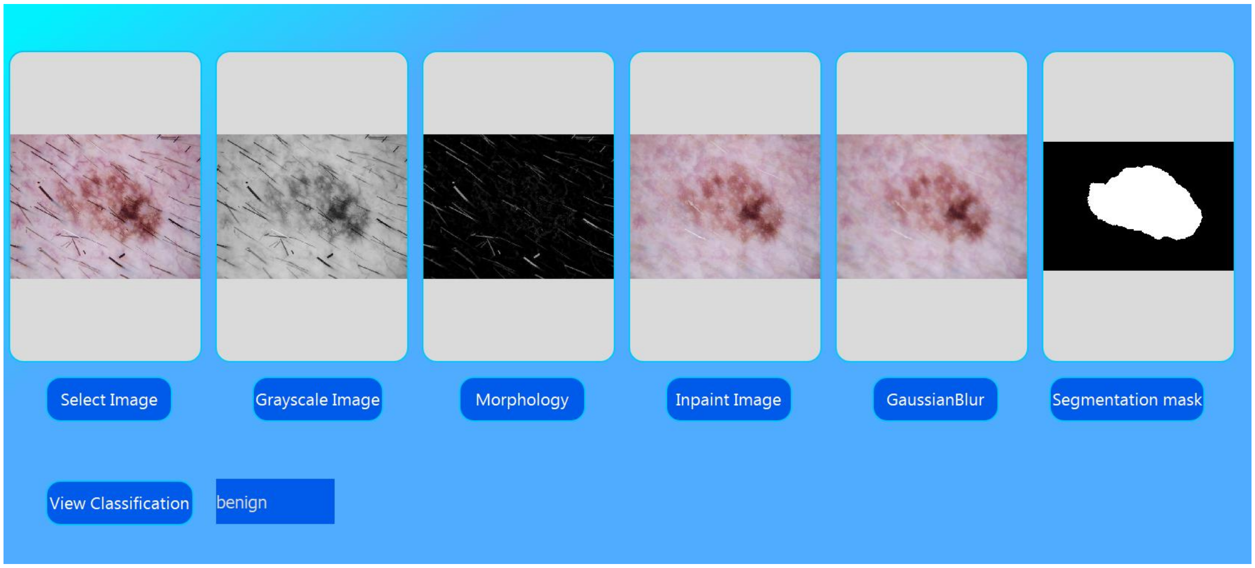
OBJECTIVE 5 RESULTS:



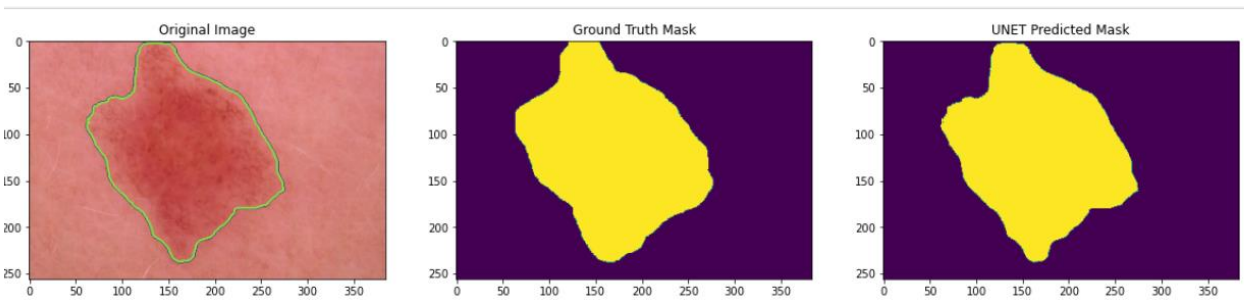
RESULTS:

All the objectives established for the project have been met and the final segmentation masks obtained using the created UNET model are very much like the ground truth masks and a moderate accuracy in skin lesion classification has been achieved by using the features derived from the ABCDT techniques.

The screenshot below shows all the sequence of steps that have been carried out in this project. The first image is the original skin lesion image, followed by the grayscale converted image. The third image shows the result of performing the morphology operation on the grayscale image. The fourth image represents the inpainted image with the thresholds calculated from the morphology operation. The fifth image represents the Gaussian blurred image which is sent as an input to our UNET model. The final image represents the segmentation map predicted by our UNET model. Lastly, clicking the “View Classification” button would show the results of the original image category.



Below is another image showing the ground truth mask against the one that has been predicted by our UNET model.



CONCLUSION:

Skin lesion identification and classification in a timely manner is a necessary component in providing the necessary treatment for the patients immediately. In this current project we have explored ways to identify the contours of a skin lesion by performing pre-processing on the input images and calculating the segmentation masks for them using UNET algorithm. Segmentation of images ensures that we identify the region of interest appropriately, which is very important in medical diagnosis. It also allows us to extract important useful information from the masks which in our case was done by using the ABCDT analysis. These features were later used to train a SVM classification algorithm to classify the skin lesion images as Benign or Cancerous. Since the final goal is to make it easier for physicians to work on these images an intuitive GUI has also been designed to visualize the segmentation masks and classification results of skin lesion images.

APPENDIX:

CODE LINK: https://github.com/MeghanaMsl/FMI-Final-Project

- *HAM10000_training.ipynb:*
  This file consists of all the python code and functions required to create the UNET segmentation model, train, test it and perform feature extraction. It also contains the code for training a SVM classifier for skin lesion classification.
- *Project_GUI.py:*
  This python file consists of the code required for the GUI implementation and involves invoking the saved model files for the UNET and SVM algorithms.
- *model-skin-lesion-segmentation-org2000.h5:*
  Represents the saved model file for the trained UNET algorithm with 2000 data points
- *svm_model_poly.pkl:*
  Represents the saved model file for the trained SVM polynomial classifier for the classification of skin lesion images.
- *Data folder:*
  Consists of the link to original HAM10000 dataset, metadata file and the empty train, test folders that will be populated with images when code is executed using HAM10000_training.ipynb file.
- *HAM10000_segmentations_lesion_tschandl folder:*
  Consists of the mask files for the HAM10000 dataset

REFERENCES:

[1] Alcón JF, Ciuhu C, Kate W, Heinrich A, Uzunbajakava N, Krekels G, et al. Automatic imaging system with decision support for inspection of pigmented skin lesions and melanoma diagnosis. IEEE J Sel Top Signal Process. 2009;3:14–25. https://ieeexplore.ieee.org/abstract/document/4786529.

[2] S. N. Hasan, M. Gezer, R. A. Azeez and S. Gülseçen, Skin Lesion Segmentation by using Deep Learning Techniques, *2019 Medical Technologies Congress (TIPTEKNO)*, 2019, pp. 1-4, doi: 10.1109/TIPTEKNO.2019.8895078.

[3] Gutman, David, et al. Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC). arXiv preprint arXiv:1605.01397 (2016).

[4] Tschandl, Philipp, 2018, The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions, https://doi.org/10.7910/DVN/DBW86T, Harvard Dataverse, V3, UNF:6:/APKSsDGVDhwPBWzsStU5A== [fileUNF].

[5] Biagiom. (n.d.). Skin-lesions-classifier/skin_lesions_classifier.ipynb at master ·BIAGIOM/skin-lesions-classifier. Retrieved December 10, 2022, from https://github.com/biagiom/skin-lesions-classifier/blob/master/skin_lesions_classifier.ipynb

[6] NET: Convolutional Networks for Biomedical Image Segmentation. (n.d.). Retrieved December 10, 2022, from https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/